ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ

# ЕГЭ-2020

Единый государственный экзамен ЕГЭ

Д.М. УШАКОВ

# ИНФОРМАТИКА

## 10 ТРЕНИРОВОЧНЫХ ВАРИАНТОВ ЭКЗАМЕНАЦИОННЫХ РАБОТ ДЛЯ ПОДГОТОВКИ К ЕДИНОМУ ГОСУДАРСТВЕННОМУ ЭКЗАМЕНУ

- ☑ НОВЫЕ типы заданий
- ☑ ВАРИАНТ с подробным разбором решений

270 ТРЕНИРОВОЧНЫХ ЗАДАНИЙ

ЕГЭ – ШКОЛЬНИКАМ И УЧИТЕЛЯМ

НОВОЕ ИЗДАНИЕ!

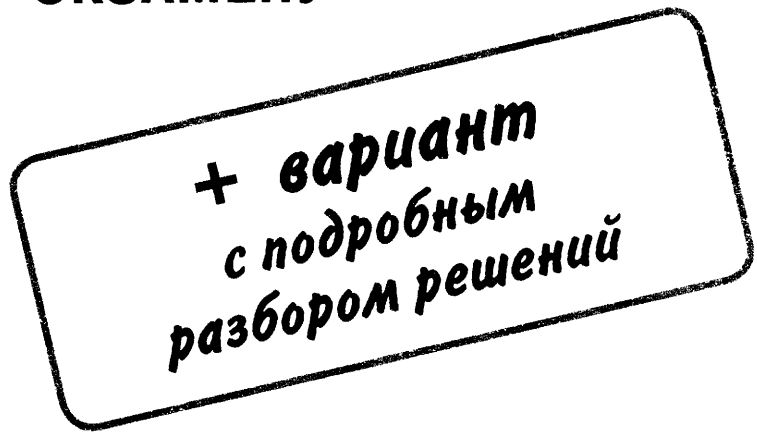
# ЕГЭ-2020

Д. М. Ушаков

# ИНФОРМАТИКА

## 10

## ТРЕНИРОВОЧНЫХ ВАРИАНТОВ ЭКЗАМЕНАЦИОННЫХ РАБОТ ДЛЯ ПОДГОТОВКИ К ЕДИНОМУ ГОСУДАРСТВЕННОМУ ЭКЗАМЕНУ




Москва
Издательство АСТ
2019

УДК 373:002
ББК 32.81я721
У93

**Ушаков, Денис Михайлович.**

У93 ЕГЭ-2020 : Информатика : 10 тренировочных вариантов экзаменационных работ для подготовки к единому государственному экзамену / Д.М. Ушаков. — Москва: АСТ, 2019. — 181, [3] с. — (ЕГЭ-2020. Это будет на экзамене).

ISBN 978-5-17-115771-5

Вниманию школьников и абитуриентов предлагается пособие для подготовки к ЕГЭ, которое содержит 10 тренировочных вариантов экзаменационных работ.

Каждый вариант составлен в соответствии с требованиями единого государственного экзамена, включает задания разных типов и сложности по основным разделам курса информатики.

В пособие включён вариант с подробным разбором решений всех заданий. Представленные алгоритмы будут полезны при проверке и оценке своих навыков и умений в решении типовых экзаменационных заданий.

В конце пособия даны ответы для самопроверки на все задания части 1 и критерии оценивания заданий с развёрнутым ответом части 2.

Материал сборника может быть использован для планомерного повторения изученного материала и тренировки выполнения заданий различного типа при подготовке к экзамену.

**УДК 373:002**
**ББК 32.81я721**

ISBN 978-5-17-115771-5

© Ушаков Д.М., 2019
© ООО «Издательство АСТ», 2019

# СОДЕРЖАНИЕ

**Предисловие** . . . . . 4
**ИНСТРУКЦИЯ ПО ВЫПОЛНЕНИЮ РАБОТЫ** . . . . . 5
**ВАРИАНТ с подробным разбором решений и ответами** . . . . . 6
**ВАРИАНТЫ ТРЕНИРОВОЧНЫХ РАБОТ** . . . . . 38
Вариант 1 . . . . . 38
Вариант 2 . . . . . 49
Вариант 3 . . . . . 60
Вариант 4 . . . . . 72
Вариант 5 . . . . . 84
Вариант 6 . . . . . 94
Вариант 7 . . . . . 106
Вариант 8 . . . . . 117
Вариант 9 . . . . . 128
Вариант 10 . . . . . 140
**ОТВЕТЫ** . . . . . 151
ОБРАЗЦЫ БЛАНКОВ ОТВЕТОВ . . . . . 182

# ПРЕДИСЛОВИЕ

Уважаемые учащиеся старших классов, абитуриенты и учителя!

Вашему вниманию предлагается сборник типовых вариантов экзаменационных работ по информатике для подготовки к ЕГЭ в 2020 году.

Содержание заданий разработано по основным темам курса информатики и ИКТ, объединённым в следующие тематические блоки: «Информация и её кодирование», «Моделирование и компьютерный эксперимент», «Системы счисления», «Логика и алгоритмы», «Элементы теории алгоритмов», «Программирование», «Архитектура компьютеров и компьютерных сетей», «Обработка числовой информации», «Технологии поиска и хранения информации».

В начале пособия включён вариант с подробным разбором решений всех заданий. Представленные алгоритмы будут полезны при проверке и оценке своих навыков и умений в решении типовых экзаменационных заданий. Этот материал будет «палочкой-выручалочкой» при возникших сложностях в работе с тем или иным заданием!

Данный сборник содержит 10 тренировочных вариантов экзаменационных работ, составленных в соответствии с демонстрационным вариантом и спецификацией 2019 года.

Каждый вариант состоит из двух частей и включает в себя 27 заданий, различающихся формой и уровнем сложности.

Часть 1 содержит 23 задания с кратким ответом нескольких разновидностей:

- задания на выбор и запись одного или нескольких правильных ответов из предложенного перечня ответов;
- задания на вычисление определённой величины;
- задания на установление правильной последовательности, представленной в виде строки символов по определённому алгоритму.

В части 1 12 заданий относится к базовому уровню, 10 заданий к повышенному уровню сложности, 1 задание — к высокому уровню сложности.

Задания проверяют материал всех тематических блоков.

Часть 2 содержит 4 задания, первое из которых повышенного уровня сложности, остальные 3 задания высокого уровня сложности. Задания этой части подразумевают запись развёрнутого ответа в произвольной форме.

Задания части 2 направлены на проверку сформированности важнейших умений записи и анализа алгоритмов и умения по теме «Технология программирования», предусмотренных требованиями к обязательному уровню подготовки по информатике учащихся средних общеобразовательных учреждений.

В конце пособия представлены ответы для проверки решений 10 тренировочных вариантов.

Если при решении заданий из данного пособия Вы обнаружите какие-то неточности или опечатки, то на странице авторского сайта www.dmushakov.ru можно посмотреть самые последние сведения о замеченных ошибках. Там же можно задать вопрос автору сборника и посмотреть, какие дополнительные пособия Д. М. Ушакова по информатике, издаваемые в нашем издательстве, могут быть Вам полезны при подготовке к экзамену.

**В связи с возможными изменениями в формате и количестве заданий рекомендуем в процессе подготовки к экзамену обращаться к материалам сайта официального разработчика экзаменационных заданий — Федерального института педагогических измерений: www.fipi.ru.**

# ИНСТРУКЦИЯ ПО ВЫПОЛНЕНИЮ РАБОТЫ

Экзаменационная работа состоит из двух частей, включающих в себя 27 заданий. Часть 1 содержит 23 задания с кратким ответом. Часть 2 содержит 4 задания с развёрнутым ответом.

На выполнение экзаменационной работы по информатике и ИКТ отводится 3 часа 55 минут (235 минут).

Ответы к заданиям 1–23 записываются в виде числа, последовательности букв или цифр. Ответ запишите в поле ответа в тексте работы, а затем перенесите в бланк ответов № 1.

КИМ
Ответ: 23.

Задания 24–27 требуют развёрнутого решения. В бланке ответов № 2 укажите номер задания и запишите его полное решение.

Все бланки ЕГЭ заполняются яркими чёрными чернилами. Допускается использование гелевой, или капиллярной, или перьевой ручки.

При выполнении заданий можно пользоваться черновиком. Записи в черновике, а также в тексте контрольных измерительных материалов не учитываются при оценивании работы.

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

***Желаем успеха!***

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связок (операций):

a) *отрицание* (инверсия, логическое НЕ) обозначается ¬ (например, ¬A);

b) *конъюнкция* (логическое умножение, логическое И) обозначается $\wedge$ (например, A $\wedge$ B) либо & (например, A & B);

c) *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается $\vee$ (например, A $\vee$ B) либо | (например, A | B);

d) *следование* (импликация) обозначается $\rightarrow$ (например, A $\rightarrow$ B);

e) *тождество* обозначается $\equiv$ (например, A $\equiv$ B). Выражение A $\equiv$ B истинно тогда и только тогда, когда значения A и B совпадают (либо они оба истинны, либо они оба ложны);

f) символ 1 используется для обозначения истины (истинного высказывания); символ 0 — для обозначения лжи (ложного высказывания).

2. Два логических выражения, содержащих переменные, называются *равносильными* (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения A $\rightarrow$ B и (¬A) $\vee$ B равносильны, а A $\vee$ B и A $\wedge$ B *неравносильны* (значения выражений разные, например, при A = 1, B = 0).

3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом, ¬A $\wedge$ B $\vee$ C $\wedge$ D означает то же, что и ((¬A) $\wedge$ B) $\vee$ (C $\wedge$ D).

Возможна запись A $\wedge$ B $\wedge$ C вместо (A $\wedge$ B) $\wedge$ C. То же относится и к дизъюнкции: возможна запись A $\vee$ B $\vee$ C вместо (A $\vee$ B) $\vee$ C.

4. Обозначения Мбайт и Кбайт используются в традиционном для информатики смысле — как обозначения единиц измерения, чьё соотношение с единицей «байт» выражается степенью двойки.

# ВАРИАНТ
# с подробным разбором решений и ответами

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько существует натуральных чисел $x$, для которых выполнено неравенство $346_8 < x < EA_{16}$? В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____.

Р е ш е н и е. Так как исходные числа даны в разных системах счисления, переведём их в одну систему счисления. Например, в десятичную. Для этого, пронумеруем у каждого числа разряды справа налево, начиная с нуля, а потом выпишем все цифры, умножив их на основание системы счисления, возведённую в степень, которую мы записали над цифрой:

$$346_8 = 3 \cdot 8^2 + 4 \cdot 8^1 + 6 \cdot 8^0 = 3 \cdot 64 + 4 \cdot 8 + 6 \cdot 1 = 192 + 32 + 6 = 230.$$

То же самое сделаем для второго числа. Только при этом нужно будет вспомнить, что E — это 14, а A — это 10:

$$EA_{16} = E \cdot 16^1 + A \cdot 16^0 = 14 \cdot 16 + 10 \cdot 1 = 224 + 10 = 234.$$

Между числами 230 и 234 находится 3 числа (234 – 230 + 1).

О т в е т: 3.

Для проверки этого ответа хорошо бы использовать альтернативный способ вычислений. Так как исходные системы счисления являются степенями двойки, исходные числа можно быстро перевести в двоичную систему счисления. Для этого каждую цифру восьмеричного числа нужно заменить на соответствующее ей трёхразрядное двоичное число. А каждую цифру шестнадцатеричного числа — на соответствующее ей четырёхразрядное двоичное число.

$346_8 = 011\ 100\ 110_2$

$EA_{16} = 1110\ 1010_2$

Вычтем из второго числа первое:

$$\begin{array}{l} 11101010_2 \\ \underline{-\ 11100110_2} \\ 00000100_2 \end{array}$$

Переведём результат в десятичную систему счисления: $100_2 = 1 \cdot 2^2 = 4$. Вычтем из результата единицу.

О т в е т: 3.

**2** Миша заполнял таблицу истинности функции $(\neg x \wedge \neg y) \vee (y \equiv z) \vee w$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $w$, $x$, $y$, $z$.

| Перем. 1 | Перем. 2 | Перем. 3 | Перем. 4 | Функция |
|---|---|---|---|---|
| ??? | ??? | ??? | ??? | F |
| 1 | | | 1 | 0 |
| 1 | 0 | 0 | | 0 |
| | | 1 | 1 | 0 |

В ответе напишите буквы $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала — буква, соответствующая 1-му столбцу; затем — буква, соответствующая 2-му столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Если бы функция была задана выражением $x \to y$, зависящее от двух переменных $x$ и $y$, а фрагмент таблицы имел бы вид:

| | | $x \to y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать: $yx$.

Ответ: _____.

Решение.

1. В каждой приведённой строке таблицы истинности результат логического выражения равен нулю, а при этом выражение состоит из трёх слагаемых дизъюнкции. Значит, каждое слагаемое равно нулю.

2. Третье слагаемое — просто переменная $w$. Значит, она в каждой строке должна быть равна нулю. Единственный столбец, который не содержит единиц — столбец 2. Значит, переменная 2 — $w$.

Внесём эти сведения в таблицу:

| Перем. 1 | Перем. 2 | Перем. 3 | Перем. 4 | Функция |
|---|---|---|---|---|
| ??? | w | ??? | ??? | F |
| 1 | 0 | | 1 | 0 |
| 1 | 0 | 0 | | 0 |
| | 0 | 1 | 1 | 0 |

3. Чтобы второе слагаемое ($y \equiv z$) равнялось нулю, необходимо, чтобы $y$ и $z$ были бы разными. В таблице в строке 1 совпадают числа в столбцах 1 и 4. А в строке 3 — в столбцах 3 и 4. Значит, пара переменных $y$ и $z$ не может находиться в столбцах 1,4 и 3,4. Единственный оставшийся для этой пары вариант — столбцы 1,3. Значит, в столбце 4 должна находиться оставшаяся переменная — $x$.

4. Так как переменные $y$ и $z$ должны иметь разные значения, и при этом они находятся в столбцах 1 и 3, ни в одной строчке таблицы в столбцах 1 и 3 не могут стоять одинаковые числа. Значит, в первой строке в третьем столбце и в третьей строке в первом столбце стоят нули.

Внесём эти сведения в таблицу:

| Перем. 1 | Перем. 2 | Перем. 3 | Перем. 4 | Функция |
|---|---|---|---|---|
| ??? | w | ??? | x | F |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | | 0 |
| 0 | 0 | 1 | 1 | 0 |

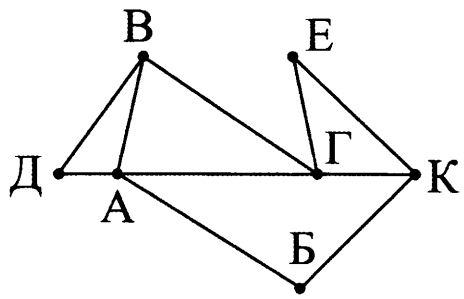
5. Так как в таблице истинности, по определению, все строки должны быть разными, в оставшейся пустой ячейке (строка 2, столбец 4) должен стоять 0 (если поставить туда 1, строки 1 и 2 будут совпадать).

6. Теперь нужно определить, в каком из столбов (1 и 3) стоит $y$, а в каком — $z$. Чтобы первое слагаемое ($\neg x \wedge \neg y$) равнялось нулю, оно не должно равняться единице. А $\neg x \wedge \neg y = 1$, если $\neg x = 1$ и $\neg y = 1$. Это происходит, если $x = 0$ и $y = 0$. То есть, переменные $x$ и $y$ не должны одновременно равняться нулю. Переменная $x$ равняется нулю только в строке 2. В этой строке в столбце 3 стоит 0, а в столбце 1 стоит 1. Значит, единственный вариант для переменной $y$ — столбец 1. Следовательно, в столбце 3 стоит $z$.

Ответ: $ywzx$.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|---|---|---|---|---|---|---|---|
| П1 | | 35 | 15 | | 55 | 20 | |
| П2 | 35 | | | 30 | 45 | | 50 |
| П3 | 15 | | | | | 10 | 25 |
| П4 | | 30 | | | 40 | | |
| П5 | 55 | 45 | | 40 | | | |
| П6 | 20 | | 10 | | | | |
| П7 | | 50 | 25 | | | | |



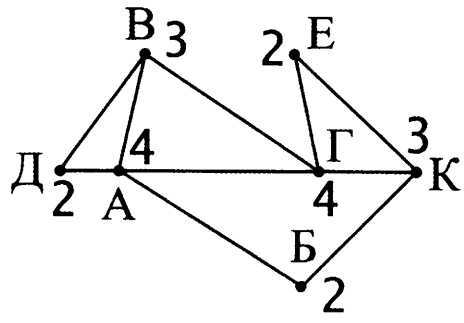

Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги и пункта Б пункт К. В ответе запишите целое число — так, как оно указано в таблице.

Ответ: _____.

Решение.

Сначала посчитаем, сколько дорог входит в каждый населённый пункт на схеме и в таблице:

| | | 4 | 4 | 3 | 2 | 3 | 2 | 2 |
|---|---|---|---|---|---|---|---|---|
| | | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
| 4 | П1 | | 35 | 15 | | 55 | 20 | |
| 4 | П2 | 35 | | | 30 | 45 | | 50 |
| 3 | П3 | 15 | | | | | 10 | 25 |
| 2 | П4 | | 30 | | | 40 | | |
| 3 | П5 | 55 | 45 | | 40 | | | |
| 2 | П6 | 20 | | 10 | | | | |
| 2 | П7 | | 50 | 25 | | | | |




Каждое полученное число дорог встречается как минимум по два раза. Поэтому быстро (просто по количеству входящих дорог) определить соответствие хотя бы одного населённого пункта мы не можем. Посмотрим, какими свойствами обладают искомые по условию пункты Б и К. В пункт Б входят 2 дороги, а в пункт К — 3. Сначала попробуем определить пункт Б (у него только две дороги, меньше дорог — легче проверять). На схеме их три. Каждый из них (Д, Е и Б) имеет соседей с количеством дорог 3 и 4. Так их не отличить друг от друга.

Тогда посмотрим на пункт К. У него 3 дороги. На схеме трёхдорожных пунктов два — В и К. При этом у пункта В соседи имеют 2, 4 и 4 дороги (Д, А и Г), а у пункта К соседи имеют 2, 4 и 2 дороги (Е, Г и Б). По этому признаку определим в таблице пункт К. В строке П3 соседи (числа 15, 10 и 25) имеют по 4, 2 и 2 дороги — подходит к нужному нам пункту К. Заметим, что в строке П5 соседи (числа 55, 45 и 40) имеют по 4, 4 и 2 дороги — не подходит.

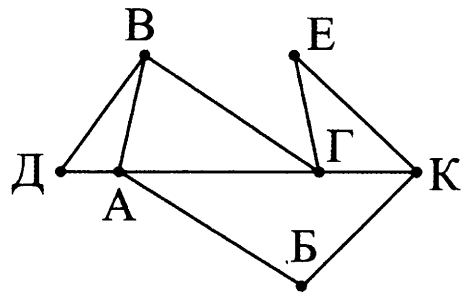
Итак, мы определили, что пункт К — это П3. Теперь нужно определить, кто из его соседей — Е, а кто — Б. По схеме видим, что пункт Е связан дорогой с тем же пунктом, что и пункт К. А пункт Б — с другим. По таблице определяем, что единственный 4-хдорожный пункт, связанный с П3 — это П1. Теперь смотрим в таблице, какой из 2-хдорожных соседей П3 (их два — П6 и П7) не связан с П1. П6 связан с П1, а П7 — нет. Значит, П7 — это Б.

На пересечении строки П3 и столбца П7 находится число 25.

Ответ: 25.

На всякий случай, приведём таблицу со всеми заполненными буквенными обозначениями населённых пунктов:

| | Г | А | К | Д | В | Е | Б |
|---|---|---|---|---|---|---|---|
| Г | | 35 | 15 | | 55 | 20 | |
| А | 35 | | | 30 | 45 | | 50 |
| К | 15 | | | | | 10 | 25 |
| Д | | 30 | | | 40 | | |
| В | 55 | 45 | | 40 | | | |
| Е | 20 | | 10 | | | | |
| Б | | 50 | 25 | | | | |




**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите на основании приведённых данных ID человека, который стал отцом в самом молодом возрасте.

*Таблица 1*

| ID | Фамилия И.О. | Пол | Год рождения |
|---|---|---|---|
| 145 | Дени К.Д. | Ж | 1957 |
| 218 | Ирта О.Б. | Ж | 1978 |
| 260 | Гано И.А. | Ж | 1980 |
| 347 | Ивенко Д.Я. | М | 1936 |
| 421 | Бабенко А.Е. | М | 1994 |
| 453 | Старых Г.В. | Ж | 1919 |
| 536 | Марченко Е.А. | М | 1975 |
| 577 | Арне А.А. | М | 1921 |
| 619 | Рудко К.А. | Ж | 2019 |
| 689 | Шевченко А.В. | М | 2002 |
| 714 | Сиро А.К. | М | 1955 |
| 781 | Исава Г.А. | Ж | 1940 |
| 820 | Сорт А.К. | Ж | 2000 |
| 907 | Смирных Н.Д. | Ж | 1960 |
| | | | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 536 | 421 |
| 218 | 421 |
| 714 | 536 |
| 145 | 536 |
| 347 | 145 |
| 453 | 781 |
| 781 | 145 |
| 714 | 260 |
| 145 | 260 |
| 347 | 907 |
| 781 | 907 |
| 260 | 820 |
| 577 | 781 |
| 689 | 619 |
| 260 | 689 |

Ответ _____.

Решение.

Так как в задаче требуется определить только самого молодого отца, достаточно рассматривать только отдельные связи «родитель-ребёнок», в которых родитель мужского пола. То есть, в отличие от многих задач подобного рода, нет необходимости строить дерево всех связей.

Будем рассматривать только связи, приведённые в таблице 2.

Чтобы не ошибиться и ничего не пропустить и не потерять, будем делать заметки прямо в приведённой таблице 2. Сначала для каждого родителя определим по таблице 1 его пол. И, если он мужской, сразу запишем рядом с ID год рождения родителя. А если пол женский, поставим прочерк (чтобы отметить, что этот элемент проверен, и не требует дальнейшего рассмотрения).

После этого, для тех строк, в которых указан год рождения родителя, найдём в таблице 1 год рождения ребёнка и запишем его рядом с ID.

*Таблица 2*

| ID_Родителя | ID_Ребёнка | |
|---|---|---|
| 536 – 1975 | 1994 – 421 | 19 |
| 218 — | 421 | |
| 714 – 1955 | 1975 – 536 | 20 |
| 145 — | 536 | |
| 347 – 1936 | 1957 – 145 | 21 |
| 453 — | 781 | |
| 781 — | 145 | |
| 714 – 1955 | 1980 – 260 | 25 |
| 145 — | 260 | |
| 347 – 1936 | 1960 – 907 | 24 |
| 781 — | 907 | |
| 260 — | 820 | |
| 577 – 1921 | 1940 – 781 | 19 |
| 689 – 2002 | 2019 – 619 | 17 |
| 260 — | 689 | |

Теперь для каждой пары годов рождения посчитаем разницу между ними и запишем её рядом:

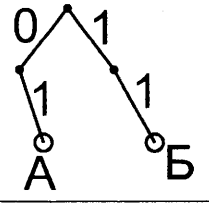
Найдём наименьшее среди полученных чисел. Это число 17. ID родителя для этой строки — 689.

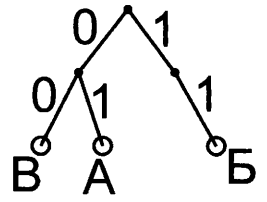
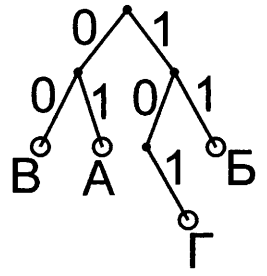
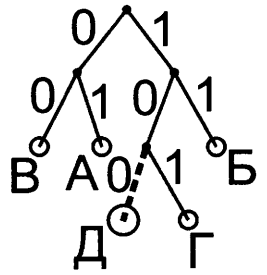
О т в е т: 689.

**5** Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, удовлетворяющий условию Фано. Вот этот код: А-01, Б-11, В-00, Г-101. Каким кодовым словом должна кодироваться буква Д? Если таких вариантов несколько, укажите самый короткий.

Ответ _____.

Р е ш е н и е. Будем строить «дерево декодирования» — двоичное дерево, при помощи которого обычно происходит декодирование неравномерных кодов, удовлетворяющих условию Фано. Также дерево декодирования очень удобно использовать при построении, а также при анализе неравномерных двоичных кодов.

| | |
|---|---|
| Будем строить двоичное дерево сверху вниз (сверху — корень, снизу — листья). Для того, чтобы ничего не забыть, будем строить дерево последовательно, дорисовывая в дерево листья для каждого по порядку кодового слова. Каждой цифре кодового слова соответствует ветка дерева декодирования. Для определённости будем рисовать ветви дерева для цифр 0 налево, а для цифр 1 — направо. Начнём с буквы А. Её код — 01. То есть, от корня первую ветку нарисуем налево, а вторую, от неё — направо. Узлы дерева отметим жирными точками, а листья (в данном случае, для буквы А) — кружками. Подпишем букву рядом с кружком. | |
| Теперь сделаем такое же действие для буквы Б. Её кодовое слово — 11. От корня дерева (который уже нарисован) нарисуем ветвь вправо (для 1), и от неё ещё одну ветвь вправо (для второй 1). Обозначим узел между ветвями (жирной точкой) и лист (кружком) для буквы Б. |  |

| | |
|---|---|
| Дорисуем дерево для буквы В. Её кодовое слово — 00. Первая ветвь — (0, влево от корня) в дереве уже нарисована (когда мы рисовали ветви для буквы А). От её конца нарисуем ещё одну ветвь влево (для второго 0). Обозначим лист для буквы В. |   |
| Дорисуем дерево для буквы Г. Её кодовое слово — 101. От корня право (для первой 1) ветвь уже есть. От её конца рисуем ветвь влево (для 0), и от неё вправо (для второй 1). Обозначаем узел дерева (жирной точкой) и лист Г (кружком). |   |
| В полученном дереве ищем, куда бы ещё можно было пририсовать ветвь. То есть, ищем узел дерева, из которого выходит только одна ветвь. Находим один такой узел. Путь к нему от корня — 10. В узле не хватает ветви влево. То есть, для 0. Значит, ветвь с листом для буквы Д можно пририсовать сюда. Она будет иметь путь от корня путь: 100. |   |

Это и есть кодовое слово для буквы Д.

Ответ: 100.

**6** На вход алгоритма подаётся натуральное число N. Алгоритм строит по нему новое число R следующим образом.

1) Строится двоичная запись числа N.

2) К этой записи дописываются разряды по следующему правилу:

а) если число чётное, то к двоичной записи числа в начало дописывается 1, а в конец дописывается 0;

б) если число нечётное, то к двоичной записи числа в начало дописывается 11, и в конец дописывается 11.

Полученная таким образом запись является двоичной записью искомого числа R.

Укажите минимальное число R, которое превышает 100 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Ответ _____.

Р е ш е н и е. Рассмотрим наименьшее число, допустимое по условию (больше 100). То есть, 101. переведём его в двоичную систему счисления: $101 = 64 + 37 = 2^6 + 32 + 5 = 2^6 + 2^5 + 4 + 1 = 2^6 + 2^5 + 2^2 + 2^0 =$ $= 1100101_2$. Рассмотрим, может ли быть оно результатом работы приведённого алгоритма. Так как число заканчивается на 1, оно не может быть результатом действия (а) (иначе число бы заканчивалось на 0). В то же время, оно не может быть результатом действия (б) (иначе число бы заканчивалось на 11).

Рассмотрим следующее число (102). Проще всего не переводить 102 в двоичную систему счисления, а прибавить к двоичному представлению числа 101 число 1. Получим $1100110_2$. Оно оканчивается на 0. Возможно, это результат действия (а). Если это так, отбросим от этого числа слева цифру 1 и справа цифру 0 (предполагаемый результат действия (а)). Получим число $10011_2$. Оно оканчивается на 1, значит, нечётное. Значит, если бы это было искомое число N, к нему нужно было бы применять действие (б), а не действие (а), как мы предположили.

Рассмотрим следующее число (103). Добавим 1 к двоичному представлению числа 102. Получим $1100111_2$. Оно начинается на 11 и заканчивается на 11. То есть, предполагаем, что оно является результатом действия (б). Отбросим левые 11 и правые 11. Получим число $0011_2$. Оно не может быть искомым числом N, так как начинается на 0, а по условию не предполагается дописывание ведущего незначащего нуля к двоичному представлению числа N.

Рассмотрим следующее число (104). Добавим 1 к двоичному представлению числа 103. Получим $1101000_2$. Оно заканчивается на 0. Предположим, что оно является результатом действия (а). Отбросим от него слева 1 и справа 0. Получим $10100_2$. Это число чётное. То есть, к нему вполне применимо действие (а). Значит, именно оно нам и нужно. Переведём это число в десятичную систему счисления:

43210

$10100_2 = 2^4 + 2^2 = 16 + 4 = 20.$

Ответ: 20.

**7** Дан фрагмент электронной таблицы.

| | A | B | C | D |
|---|---|---|---|---|
| **1** | 1 | 2 | 3 | |
| **2** | 5 | 6 | =A2+B1 | |
| **3** | 7 | 8 | =B$1+B3 | |

Чему станет равным значение ячейки D2, если в неё скопировать формулу из ячейки C3?
*Примечание: знак $ обозначает абсолютную адресацию.*

Ответ _____.

Решение. При копировании из ячейки C3 в ячейку D2 номер столбца меняется на «+1», а номер строки меняется на «−1». В формуле =B$1+B3 индекс $1 не изменится. Индекс B изменится на «+1» и станет C. Индекс 3 изменится на «−1» и станет 2. Получим формулу =C$1+C2. Вычислим её. В ячейке C2 находится формула =A2+B1. A2=5, B1=2. Значит, C2=5+2=7. C1=3. Вычислим D2=C1+C2=3+7=10.

Ответ: 10.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 1
s = 8
WHILE k < 500
  s = s + 20
  k = k * 5
WEND
PRINT s
```

**Паскаль**

```
var k, s : integer;
begin
  k := 1; s := 8;
  while k < 500 do begin
    s := s + 20;
    k := k * 5
  end;
  write(s)
end.
```

**C++**

```
#include <iostream>
int main() {
   int k = 1, s = 8;
   while (k < 500) {
     s = s + 20;
     k = k * 5;
   }
   std::cout << s;
   return 0
}
```

**Алгоритмический язык**

```
нач
  цел k, s
  k := 1
  s := 8
  нц пока k < 500
    s := s + 20
    k := k * 5
  кц
  вывод s
кон
```

**Python**

```
k = 1; s = 8
while k < 500:
    s = s + 20
    k = k * 5
print(s)
```

Ответ _____.

Р е ш е н и е.

Способ 1. Анализируем программу.

Цикл выполняется, пока k < 500. Переменная k изначально равна 1. На каждом шаге цикла значение переменной k увеличивается в 5 раз. Условие цикла перестанет выполняться при k = 625. То есть, значение переменной k увеличивалась в 5 раз четырежды. Значит, цикл выполнялся 4 раза. На каждом шаге цикла значение переменной s увеличивалось на 20. Значит, за 4 шага цикла значение переменной s увеличилось на 4 · 20 = 80. Начальное значение переменной s было равно 8. Значит, итоговое значение переменной s станет равно 8 + 80 = 88.

О т в е т: 88.

Способ 2. Выполним программу по шагам (выполним трассировку программы).

Рассмотрим подробную сортировку данной программы на языке Паскаль.

| Оператор | Условие | Переменные | | Пояснения |
|---|---|---|---|---|
| | | k | s | |
| k := 1; | | 1 | | Переменная k получает начальное значение, равное единице. |
| s := 8; | | | 8 | Переменная s получает начальное значение, равное 8. |
| while k < 500 do | 1<500? Да | | | Начинается цикл с предусловием while. Так как условие выполняется, выполняем тело цикла. |
| s := s + 20; | | | 28 | Значение переменной s (8) увеличивается на 20. |
| k := k * 5 | | 5 | | Значение переменной k (1) увеличивается в 5 раз. |
| while k < 500 do | 5<500? Да | | | Снова проверяется условие цикла while. Так как условие выполняется, выполняем тело цикла. |
| s := s + 20; | | | 48 | Значение переменной s (28) увеличивается на 20. |
| k := k * 5 | | 25 | | Значение переменной k (5) увеличивается в 5 раз. |
| while k < 500 do | 25<500? Да | | | Снова проверяется условие цикла while. Так как условие выполняется, выполняем тело цикла. |
| s := s + 20; | | | 68 | Значение переменной s (48) увеличивается на 20. |
| k := k * 5 | | 125 | | Значение переменной k (25) увеличивается в 5 раз. |
| while k < 500 do | 125<500? Да | | | Снова проверяется условие цикла while. Так как условие выполняется, выполняем тело цикла. |
| s := s + 20; | | | 88 | Значение переменной s (68) увеличивается на 20. |
| k := k * 5 | | 625 | | Значение переменной k (125) увеличивается в 5 раз. |
| while k < 500 do | 625<500? Нет | | | Снова проверяется условие цикла while. Так как условие не выполняется, цикл заканчивается. |
| write(s) | | | | На экране — текущее значение переменной s (88). |

О т в е т: 88.

**9** Автоматическая фотокамера производит растровые изображения размером 2400 на 1600 пикселей. При этом объём файла с изображением не может превышать 2500 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре изображения?

Ответ _____.

Решение.

Воспользуемся формулой для вычисления объёма растрового изображения:
$V = H \cdot W \cdot i$, где:
$V$ — объём изображения (в битах),
$H$ и $W$ — высота и ширина изображения (в пикселях),
$i$ — число бит, которое тратится на хранение одного пикселя изображения.
Переведём 2500 Кбайт в биты. 1 Кбайт = 1024 байта. А 1 байт = 8 бит.
Значит, 1 Кбайт = $8 \cdot 1024$ бита = $2^3 \cdot 2^{10} = 2^{13}$ бит.
Подставим все известные величины в формулу. Так как по условию объём не может превышать 2500 Кбайт, будем использовать знак $\geqslant$ вместо знака равно.
$2500 \cdot 2^{13} \geqslant 2400 \cdot 1600 \cdot i \Rightarrow$ (поделим обе части на 25 и на 100) $2^{13} \geqslant 24 \cdot 4 \cdot 16 \cdot i \Rightarrow$
$\Rightarrow 2^{13} \geqslant 3 \cdot 8 \cdot 4 \cdot 16 \cdot i \Rightarrow 2^{13} \geqslant 3 \cdot 2^3 \cdot 2^2 \cdot 2^4 \cdot i \Rightarrow 2^{13} \geqslant 3 \cdot 2^9 \cdot i \Rightarrow$ (поделим обе части на $2^9$) $2^4 \geqslant 3 \cdot i \Rightarrow$
$\Rightarrow i \leqslant 16/3 \approx 5{,}33$. Так как $i$ — целое число, наибольшее $i = 5$. Отсюда найдём наибольшее возможное число цветов в палитре изображения (по формуле $2^{\text{число бит}} =$ число цветов). $2^5 = 32$.

Ответ: 32.

**10** Все 4-буквенные слова, составленные из букв И, А, О, Е, записаны в алфавитном порядке. Вот начало списка:

**1. АААА**
**2. АААЕ**
**3. АААИ**
**4. АААО**
**5. ААЕА**

Под каким номером в списке идёт первое слово, которое начинается в буквы И?

Ответ _____.

Решение.

Данный список — это запись последовательных чисел в четверичной системе счисления. Четверичной — потому что используются 4 различных символа. Запишем эти символы в алфавитном порядке: А, Е, И, О (можно посмотреть на последние символы в строках 1–4). Для каждого из этих символов запишем соответствующую ему четверичную цифру (от 0 до 3): А – 0, Е – 1, И – 2, О – 3.

То есть, в первой строке записано четверичное четырёхразрядное число 0000, во второй строке — 0001, и т.д.

Самое первое слово списка, которое начинается с буквы И — это самое маленькое четырёхразрядное число, которое имеет первую цифру 2. То есть, число $2000_4$. Переведём это число из четверичной системы счисления в десятичную. $2000_4 = 2 \cdot 4^3 = 2 \cdot 64 = 128$.
3210

Однако, это ещё не ответ. Как мы ранее заметили, в строке номер 1 списка записано число 0000. То есть, 0. В строке номер 2 списка записано число 0001. То есть, 1. Нетрудно видеть, что число, записанное в строке, на 1 меньше, чем номер строки. Соответственно, номер строки на 1 больше, чем число, которое в этой строке записано. Значит, число 128 будет записано в строке номер 129.

Ответ: 129.

**11** Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов и разделителей), которые будут напечатаны на экране при выполнении вызова F(4).

**Алгоритмический язык**

```
алг F(цел n)
нач
  вывод n
  если n > 1 то
    F(n - 1)
    F(n - 2)
  все
кон
```

**Паскаль**

```
procedure F(n: integer);
begin
  write(n);
  if n > 1 then
  begin
    F(n - 1);
    F(n - 2)
  end
end;
```

**C++**

```
void F(int n) {
  std::cout << n;
  if (n > 1) {
    F(n - 1);
    F(n - 2);
  }
}
```

**Бейсик**

```
SUB F(n)
  PRINT n;
  IF n > 1 THEN
    F(n - 1)
    F(n - 2)
  END IF
END SUB
```

**Python**

```
def F(n):
    print(n)
    if n > 1:
        F(n - 1)
        F(n - 2)
```

Ответ _____.

Решение.

Выполним вызовы F($n$) последовательно, начиная с начального значения $n$ до F(4). Определим начальное значение $n$. Рекуррентные вызовы F($n$–1) и F($n$–2) делаются для наименьшего $n > 1$. То есть, для $n = 2$ выполняются F(2–1) = F(1) и F(2–2) = F(0). Значит, начнём с F(0). В столбце F($n$) будем записывать, что выводит на экран F($n$).

| $n$ | F($n$) | Пояснение |
|---|---|---|
| 0 | 0 | Для $n = 0$ не выполняется условие $n > 1$. Поэтому единственное, что делает F($n$), это выводит на экран $n$ (то есть, 0) |
| 1 | 1 | Аналогично F(0), F(1) только выводит на экран $n$ (то есть, 1) |
| 2 | 210 | При $n = 2$ условие $n > 1$ выполняется. Поэтому F(2) сначала выводит на экран 2, а потом вызывает F(2–1) = F(1) и F(2–2) = F(0). То, что выводят F(1) и F(0), мы уже определили ранее. Эти цифры выведутся **после** цифры 2, друг за другом, потому что F(1) и F(0) вызываются **после** вывода цифры 2, друг за другом. |
| 3 | 32101 | Условие $n > 1$ снова выполняется. Выводим на экран 3, а потом вызываем F(3–1) = = F(2) и F(3–2) = F(1). Дописываем их результаты справа от цифры 3. |
| 4 | 432101210 | Условие выполняется. Выводим 4. Вызываем F(4–1) = F(3) и F(4–2) = F(2). Дописываем их результаты справа от 4. |

Ответ: 432101210.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

*Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.*

Для узла с IP-адресом 231.117.87.213 адрес сети равен 231.117.87.208. Чему равно наибольшее возможное количество нулей в последнем байте маски сети?

Ответ _____.

Р е ш е н и е. Заметим, что при преобразовании IP-адреса в адрес сети первые три байта остались неизменными, а последний байт из 213 превратился в 208. Переведём оба числа (213 и 208) в двоичную систему счисления. $213 = 11010101_2$, $208 = 11010000_2$. Запишем эти числа друг под другом, чтобы проанализировать возможную маску при поразрядной конъюнкции:

```
 11010101
&abcdefgh
 11010000
```

Заметим, что разряды a, b и d как были равны единице, так и остались ими. Значит, a = 1, b = 1, d = 1. Разряды f и h были единицами, а стали нулями. Значит, f = 0, h = 0. Получаем текущий четвёртый байт маски: 11c1e0g0. Вспоминая, что «в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули», делаем вывод, что c = 1, g = 0. Так как по условию требуется найти маску с наибольшим количеством нулей, оставшийся неизвестный разряд маски принимаем равным нулю. Получаем, что последний байт маски: 11110000. В нём 4 нуля.

О т в е т: 4.

**13** При регистрации в компьютерной системе каждому пользователю выдается пароль, состоящий из 15 символов. Из соображений информационной безопасности каждый пароль должен содержать хотя бы 1 десятичную цифру, а также как прописные, так и строчные латинские буквы. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей.

Для хранения сведений о 15 пользователях потребовалось 600 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

Ответ _____.

Р е ш е н и е. Анализируя задачу, определяем, что сначала кодируется каждый символ минимально возможным целым количеством бит. По формуле $2^i \geqslant N$ определим количество бит, которым кодируется каждый символ. Количество вариантов каждого символа — 10 (число различных десятичных цифр), плюс 26 (число букв латинского алфавита), плюс ещё 26 (так как буквы используются как строчные, так и прописные). Получаем $N = 10 + 26 + 26 = 62$. Решаем неравенство: $2^i \geqslant 62$. Получаем наименьшее $i = 6$ бит на каждый символ. Всего в пароле 15 символов. Значит, для записи пароля требуется $15 \cdot 6 = 90$ бит. По условию, пароль кодируется минимально возможным количеством байт. Переведём 90 бит в байты. Поделим для этого 90 на 8. Получаем 11 целых и 2/8. Минимальное количество байт, в которое это поместится — 12.

Так как для 15-ти пользователей потребовалось 600 байт, то для одного пользователя требуется 600/15 = 40 байт. Из них, как бы только что подсчитали, 12 байт требуется для хранения собственно пароля. Значит, для дополнительных сведений требуется 40 – 12 = 28 байт.

Ответ: 28.

**14** Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

**А) заменить (v, w).**

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w. Например, выполнение команды

**заменить (222, 34)**

преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v, то выполнение команды **заменить (v, w)** не меняет эту строку.

**Б) нашлось (v).**

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА **условие**

  **последовательность команд**

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ **условие**

  ТО **команда1**

  ИНАЧЕ **команда2**

КОНЕЦ ЕСЛИ

выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 64 идущих подряд цифр 5? В ответе запишите полученную строку.

```
НАЧАЛО
ПОКА нашлось (444) ИЛИ нашлось (555)
  ЕСЛИ нашлось (444)
    ТО заменить (444, 5)
    ИНАЧЕ заменить (555, 4)
  КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```

Ответ _____.

Решение.

Этап 1. Анализируем программу. Понимаем, что до тех пор, пока в строке имеется комбинация «444», в первую очередь эта комбинация будет заменяться на «5». В исходной строке 64 цифры 5. То есть, в ней изначально нет ни одной цифры 4. Это значит, что на первых шагах программы будут срабатывать случаи «ИНАЧЕ».

Этап 2. Так как в строке теперь нет «444», то первые «555» заменятся на «4».

После этого в строке всё также нет «444». Поэтому самые левые «555» (они стоят сразу после первой цифры 4) заменятся на «4». В строке теперь первые два символа — «44», а затем 58 пятерок.

В строке всё также нет «444». Поэтому самые левые «555» (они стоят сразу после цифр «44») заменятся на «4». Но после этого в начале строки образовалось «444». Так как условие **«нашлось** (444)» проверяется в программе в первую очередь, то выполнится команда **«заменить** (444, 5)». То есть, эти первые «444» превратятся в цифру 5.

Этап 3. Замечаем, что в результате Этапа 2 первые девять цифр 5 заменились на одну цифру 5. То есть, из девяти цифр 5 в строке пропало восемь цифр 5. Будем по очереди «убирать» из строки по восемь цифр 5, если в строке есть хотя бы девять цифр 5.

Из 64-х цифр 5 останется 56.

Из 56-ти цифр 5 останется 48.

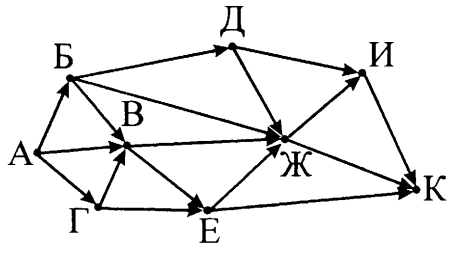
Из 48 — 40. Из 40 — 32. Из 32 — 24. Из 24 — 16. Из 16 — 8.

Теперь нельзя сказать, что эти восемь цифр 5 пропадут, потому что это происходит только при хотя бы девяти цифрах 5. А их осталось только 8. Применим к этим цифрам программу, без упрощений.

Самые левые «555» заменятся на «4». Получится «455555». Теперь самые левые «555» заменятся на «4». Получится «4455». В этой строке нет ни одной комбинации «444» или «555», поэтому программа на этом закончится.

Ответ: 4455.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И и К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ _____.

Решение.

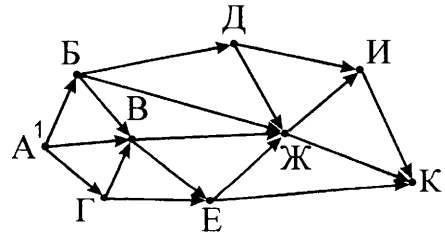
*Замечание.* Традиционный на первый взгляд метод решения этой задачи — водить пальцем по схеме от города к городу и подсчитывать, сколькими различными способами это можно сделать. Этот способ по надежности не выдерживает никакой критики — пропустить один или несколько вариантов очень легко, а проверить себя при этом практически невозможно.

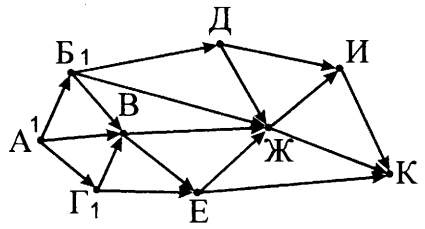
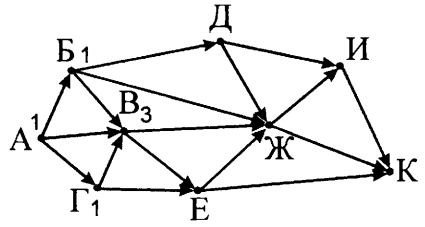
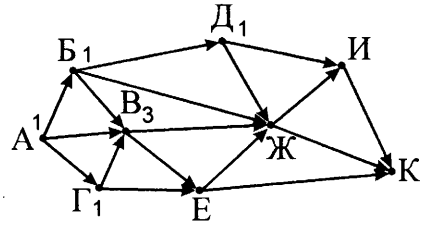
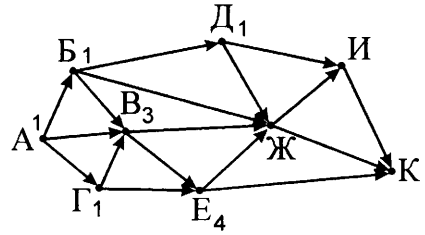
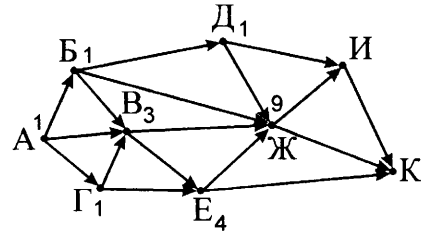
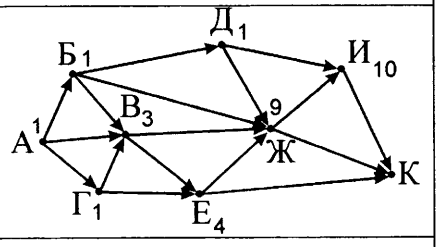
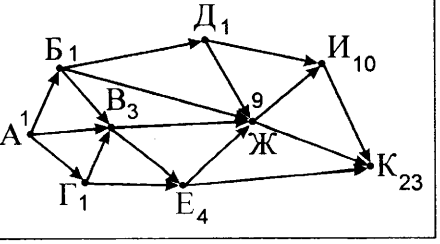
Будем использовать для решения быстрый и эффективный способ решения под названием *динамическое программирование*. Пусть Вас не смущает слово программирование. Никакого программирования при этом не будет. Просто этот способ используется в программировании при решении ряда подобных или похожих по принципу решения задач. Идея метода динамического программирования состоит в том, чтобы постепенно, шаг за шагом, получать частичные результаты для каждой рассматриваемой вершины. В результате работы метода на последнем шаге мы получим ответ для искомой вершины. В данном случае интересующий нас результат — это сколькими различными способами можно «доехать» из стартового города до текущей вершины (города).

Будем последовательно искать, сколькими различными способами можно добраться из города **А** до каждого из городов, нарисованных на схеме.

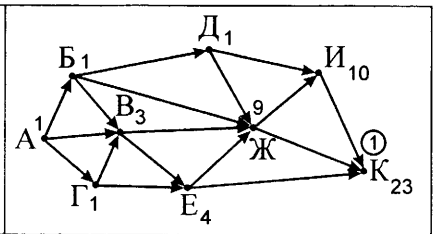
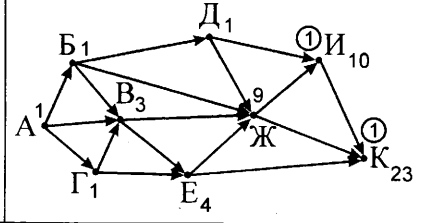
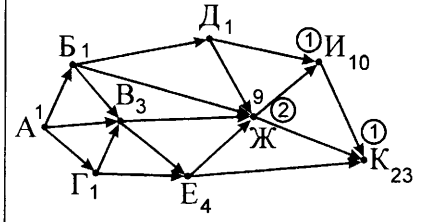
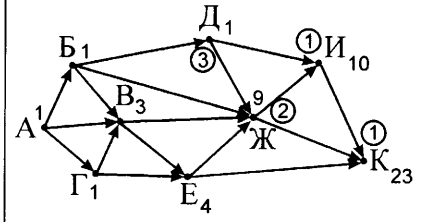
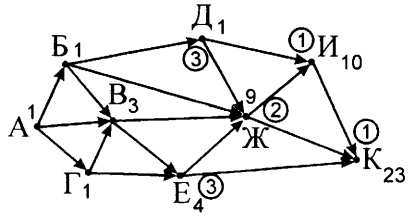
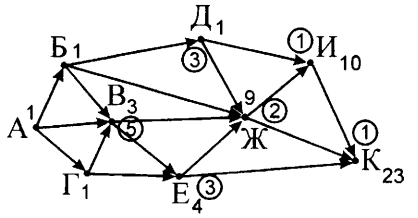
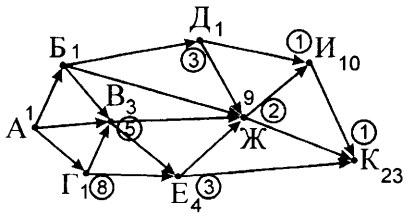
Прежде всего, для выполнения нужного нам решения необходимо понять, сколькими способами можно добраться из города **А** до самого города **А**. Важно понимать, что это можно сделать одним способом. Не нулём, как могло бы показаться поначалу. А именно одним способом — никуда не ездя.

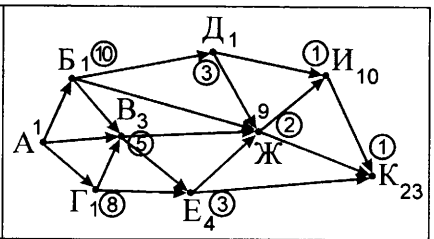
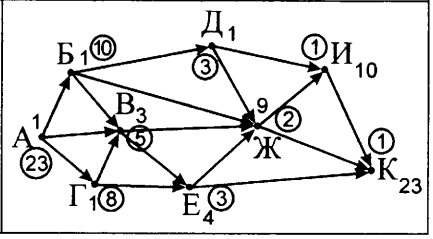
Пометим вершину **А** числом 1 (количество способов, которым можно добраться до данной вершины).



| | |
|---|---|
| Теперь постараемся пометить какую-нибудь ещё вершину. Это можно сделать только для тех вершин, про которые известно нужное количество для всех вершин, входящих в данную вершину. Мы на данном этапе знаем нужное количество только для одной вершины — **А**. Значит, будем искать вершины, в которые входит только одна стрелка дороги — из вершины **А**. Таких городов на данной схеме два — **Б** и **Г**. Пометим их числами 1. Это количество способов добраться из города **А** в города **Б** и **Г**. |   |
| Опять будем искать город, про который для всех стрелок, входящих в него, на схеме уже написаны числа. Один из таких городов — город **В**. В него входит три стрелки. Для каждой из стрелок (из городов **А**, **Б**, **Г**) уже написаны числа. Нужное число для текущего города **В** — это сумма чисел на начальных сторонах стрелок. То есть, в город **В** можно придти только по трём дорогам-стрелкам и города **А**, **Б** или **Г**. В каждый из них можно придти одним способом. Значит, в город **В** можно придти тремя способами. Пометим город **В** числом 3. |   |
| Найдём следующий город, для которого известны числа на началах всех входящих в него стрелках. Один из таких городов — **Д**. В него входит только одна стрелка из города **Б**. Значит, в город **Д** можно добраться столькими же способами — одним. Пометим город Д числом 1. |   |
| Найдём следующий город, про который известны числа из всех сходящих стрелок. На данном этапе это только один город — **Е**. В него входит две стрелки-дороги (**В** и **Г**). Из города **В** — 3 способа, из города **Г** — 1 способ. Всего 4 способа. Пометим город **Е** числом 4. |   |
| Ищем следующий город. На данном этапе это город **Ж**. В него входит 4 дороги. Про все эти стрелки-дороги известны числа для городов, из которых исходят стрелки. Это города **Д**, **Б**, **В**, **Е**. Рядом с ними написаны числа 1, 1, 3, 4. Сумма чисел — 9. Это количество способов (9) подписываем рядом с городом **Ж**. |   |
| Ищем следующий город. Это город **И**. В него входит две дороги — из города **Д** и города **Ж**. Рядом с ними написаны числа 1 и 9. Их сумма — 10. Это количество напишем рядом с городом **И**. |   |
| Последний город, для которого осталось посчитать нужное количество способов — город **К**. В него входит три стрелки-дороги (из городов **И**, **Ж**, **Е**). Рядом с ними написаны числа 10, 9, 4. Их сумма — 23. Это количество пишем рядом с городом **К**. Это и есть ответ. |   |

Приведённый алгоритм простой и эффективный. Однако при его выполнении можно легко сделать ошибку в арифметических вычислениях или случайно пропустить какую-нибудь стрелку, входящую в вершину. Необходимо сделать проверку. Выполним этот же алгоритм в обратную сторону. То есть, теперь будем искать количество способов, которым можно добраться из каждого города, обозначенного на схеме, до города **К**.

| | |
|---|---|
| Прежде всего, обозначим количество способов, которым можно добраться из вершины **К** в вершину **К**. Как и ранее, это количество равно 1. Подпишем это число (1) возле города **К**. Но, чтобы не перепутать числа обратного алгоритма с числами прямого алгоритма, будем обводить кружком числа обратного алгоритма. |  |
| В прямом алгоритме мы искали все города, для которых были известны числа на началах всех входящих стрелок. Теперь мы будем искать города, для которых известны числа для всех исходящих стрелок.<br>На данном шаге нас устроят только города, из которых выходит только одна стрелка — в город **К**. Такой город один — город **И**. Подпишем рядом с ним число 1 в кружке. |  |
| Теперь будем искать города, из которых все исходящие стрелки ведут в города с кружочками. Такая вершина одна — **Ж**. Из неё идут две стрелки — в города **И** и **К**. Рядом с ними в кружках написаны числа 1 и 1. Их сумма — 2. Подпишем рядом с вершиной **Ж** число 2 в кружке. |  |
| Ищем следующий город. Один из городов, у которого каждая исходящая стрелка приходит в город с кружком — город **Д**. Из него выходит две стрелки (в города **Ж** и **И**). Числа на концах стрелок — 2 и 1. Их сумма — 3. Пишем число 3 в кружке рядом с городом **Д**. |  |
| Следующий город — город **Е**. Из него выходит две стрелки (в города **Ж** и **К**). Числа в кружках на концах стрелок — 2 и 1. Их сумма — 3. Пишем число 3 в кружке рядом с городом **Е**. |  |
| Теперь можно посчитать число вариантов для города **В**. Из него выходит две стрелки (в города **Е** и **Ж**). Числа в кружках на концах стрелок — 3 и 2. Их сумма — 5. Пишем число 5 в кружке возле города **В**. |  |
| Теперь можно посчитать число вариантов для городов **Б** и **Г**. Начнём, например, с города **Г**. Из него выходит две стрелки (в города **В** и **Е**). Числа в кружках на концах стрелок — 5 и 3. Их сумма — 8. Пишем число 8 в кружке возле города **Г**. |  |

| | |
|---|---|
| Теперь посчитаем число вариантов для города **Б**. Из него выходит три стрелки-дороги (в города **В**, **Ж**, **Д**). Числа в кружках на концах стрелок — 5, 2, 3. Их сумма — 10. Пишем число 10 в кружке возле города **Б**. |  |
| Наконец, можно посчитать число вариантов для города **А**. Из него выходит три стрелки-дороги (в города **Б**, **В**, **Г**). Числа в кружках на концах стрелок — 10, 5, 8. Их сумма — 23. Пишем это число в кружке возле города **А**. Это ответ.<br>Результаты прямого и обратного алгоритма совпали. Значит, этот ответ — правильный. |  |

Мы настоятельно рекомендуем при решении этой задачи обязательно выполнять оба алгоритма — прямой и обратный. И сверять результаты их работы. Если результаты не совпадут — искать ошибку в обоих алгоритмах. Начинать искать ошибку рекомендуем с того алгоритма, результат которого дал меньшее число. Эта рекомендация дана на основании наиболее частой ошибки — забыть добавить в сумму какое-нибудь из чисел или не заметить одну из стрелок. Поэтому, как правило, меньший результат — неверный. Это не значит, что нужно сразу принять за верный ответ большее из двух результатов. Это просто рекомендация начать поиск ошибки с меньшего результата. Для принятия решения о том, что найденный ответ — верный, результаты двух алгоритмов должны совпасть.

**Замечание.** В этой задаче иногда может появиться дополнительное условие вида «пути из начального города в конечный город должны обязательно пройти через город такой-то». В этом случае нужно вычеркнуть на схеме все дороги, которые позволяют «объехать» указанный город. Для проверки того, что вы сделали вычеркивание правильно, убедитесь, что схема при этом превратилась в соединение двух схем — до указанного города, и после него. Для полученной схемы примените тот же алгоритм, который был только что подробно описан в приведённых таблицах. Обязательно, для проверки, примените алгоритм в обе стороны!

**16** Значение арифметического выражения: $9^{10} + 3^{23} - 9$ — записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

Ответ _____.

Р е ш е н и е. Представим искомое выражение в виде суммы и разности степеней тройки. $9^{10} = 3^{20}$. $9 = 3^2$. Запишем полученные слагаемые в порядке убывания степеней: $3^{23} + 3^{20} - 3^2$.

Исследуем, что за троичное число получится, если сложить $3^x$ и $3^y$ ($x \neq y$). Рассмотрим это на примере $3^8 + 3^3$. $3^8 = 100000000_3$, $3^3 = 1000_3$. Найдём их сумму (столбиком):

```
 100000000
+    1000
-----------
 100001000
```

Видим, что при сложении степеней тройки в троичной системе получается число, состоящее их двух единиц, стоящих в соответствующих позициях.

Теперь исследуем, что за троичное число получится, если вычесть $3^x$ и $3^y$ ($x > y$). Рассмотрим это на примере $3^8 - 3^3$. $3^8 = 100000000_3$, $3^3 = 1000_3$. Найдём их разность (столбиком):

```
 100000000
-    1000
-----------
  22222000
```

Видим, что при вычислении $3^x - 3^y$ в троичной системе получается число, состоящее их $(x - y)$ двоек, а затем из $y$ нулей.

Рассмотрим наше выражение $3^{23} + 3^{20} - 3^2$. Видим в нём разность степеней $3^{20} - 3^2$, к которой слева прибавляется степень числа 3. Количество двоек в троичном представлении $3^{20} - 3^2$ равно $20 - 2 = 18$. При добавлении к этому числу $3^{23}$ в троичном представлении получается ещё одна единица в позиции номер 23. Она не влияет на количество двоек в ответе.

Ответ: 18.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ « | », а для логической операции «И» — «&». В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| Колбаса | 11 |
| Паштет | 12 |
| Сыр | 15 |
| Сыр & Паштет | 5 |
| Колбаса & Сыр | 4 |
| Колбаса & Паштет | 0 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу:
**Колбаса | Сыр | Паштет ?**
Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ _____.

Решение.

Воспользуемся формулой «включений и исключений»:
К | С | П = К + С + П – К&С – К&П – С&П + К&С&П
В ней для вычисления количества объектов в объединении трёх множеств предлагается сложить количество элементов во всех трёх множествах. Из полученного количества вычесть всё количество элементов во всех возможных попарных пересечениях множеств (потому что эти элементы были добавлены в первую сумму по два раза). А затем к результату прибавить количество элементов в пересечении всех трёх множеств (потому что в результате предыдущих действий эти элементы были трижды добавлены в сумму, а потом трижды вычтены из неё).

На первый взгляд кажется, что для использования в данной задаче приведённой формулы не хватает значения одного из слагаемых (К&С&П).

Однако, если заметить, что пересечение множеств **Колбаса & Паштет** имеет 0 элементов, можно сделать вывод, что пересечение всех трёх множеств также имеет 0 элементов. Подставляем в формулу все имеющиеся значения:
К | С | П = 11 + 12 + 15 – 4 – 0 – 5 + 0 = 38 – 9 = 29.

Ответ: 29.

**18** Укажите наибольшее целое неотрицательное значение $A$, при котором выражение

$(y < x) \vee (A < y) \vee (2x + 3y < 140)$

истинно для любых целых неотрицательных значений $x$ и $y$.

Ответ _____.

Решение. Анализируя исходное выражение, видим, что оно истинно при выполнении хотя бы одного условия: $(y < x)$ или $(2x + 3y < 140)$. Значит, для значений $x$ и $y$, которые не удовлетворяют это-

му условию, значение выражения зависит исключительно от условия ($A < y$). Найдём, при каких $x$ и $y$ условие $(y < x) \vee (2x + 3y < 140)$ ложно. Для этого применим операцию отрицания к этому выражению: $\neg((y < x) \vee (2x + 3y < 140))$. По закону де Моргана это равно: $\neg(y < x) \wedge \neg(2x + 3y < 140)$. Внесём отрицания в скобки и инвертируем операции сравнения. Получаем: $(y \geqslant x) \wedge (2x + 3y \geqslant 140)$.

Нам нужно найти ограничения на $y$ (так как, по условию, нам нужно найти $A$, а оно находится в выражении $(A < y)$). Из выражения $(y \geqslant x)$, подставим ограничение для $x$ в выражение $2x + 3y$: $2x + 3y \leqslant 2y + 3y = 5y$. В то же время, выражение $2x + 3y \geqslant 140$. Получаем: $140 \leqslant 2x + 3y \leqslant 5y$. Отсюда находим, что $140 \leqslant 5y$. Разделим обе части неравенства на 5: $28 \leqslant y$. То есть, наименьшее значение $y$ — это 28. Значит, чтобы условие $(A < y)$ выполнялось всегда, нужно чтобы значение A было меньше 28. Значит, наибольшее требуемое значение $A$ равно 27.

Ответ: 27.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 10; 7; 9; 11; 7; 8 соответственно, т.е. A[0] = 4; A[1] = 5 и т.д. Определите значение переменной $k$ после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
n = 3
FOR i = 0 TO 9
  IF A(i) >= A(n) THEN
    k = k + A(i) MOD A(n)
    t = A(i)
    A(i) = A(n)
    A(n) = t
  END IF
NEXT i
```

**Алгоритмический язык**

```
k := 0
n := 3
нц для i от 0 до 9
  если A[i] >= A[n] то
      k := k + mod(A[i],A[n])
      t := A[i]
      A[i] := A[n]
      A[n] := t
  все
кц
```

**C++**

```
k = 0;
n = 3;
for (i = 0 ; i <= 9 ; i++)
  if (A[i] >= A[n]) {
    k = k + A[i] % A[n];
    t = A[i];
    A[i] = A[n];
    A[n] = t;
  }
```

**Паскаль**

```
k := 0;
n := 3;
for i := 0 to 9 do
  if A[i] >= A[n] then
  begin
    k := k + A[i] mod A[n];
    t := A[i];
    A[i] := A[n];
    A[n] := t
  end;
```

**Python**

```
k = 0
n = 3
for i in range(10):
    if A[i] >= A[n]:
        k = k + A[i] % A[n]
        t = A[i]
        A[i] = A[n]
        A[n] = t
```

Ответ _____.

Решение.

Анализируем программу. В ней перебираются все элементы массива (от 0 до 9) и каждый из них сравнивается с текущим значением элемента A[n]. Где переменная n не меняется и всегда равна 3. То есть, каждый раз сравнивается с элементом массива номер 3.

Если текущий элемент оказывается больше или равен элемента A[3], то:

- К переменной *k* прибавляется значение остатка от деления текущего элемента (A[i]) на A[3].
- После этого значения элементов A[i] и A[3] меняются местами.

Выполним программу по шагам. Будем записывать изменения значений переменных в таблице трассировки.

| Действие | Переменные | | | | | | | | | | | | Примечание |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *i* | *k* | A[0] | A[1] | A[2] | A[3] | A[4] | A[5] | A[6] | A[7] | A[8] | A[9] | |
| Начало | | 0 | 4 | 5 | 4 | 7 | 10 | 7 | 9 | 11 | 7 | 8 | |
| A0≥A3? | 0 | | | | | | | | | | | | 4≥7? Нет |
| A1≥A3? | 1 | | | | | | | | | | | | 5≥7? Нет |
| A2≥ A3? | 2 | | | | | | | | | | | | 4≥7? Нет |
| A3≥ A3? | 3 | | | | | | | | | | | | 7≥7? Да |
| *k*=*k*+... | | 0 | | | | | | | | | | | |
| В ячейку *k* добавляется остаток от деления A[3] на A[3]. Так как они равны, остаток от деления равен нулю, и значение переменной *k* не меняется. | | | | | | | | | | | | | |
| Обмен A3 и A3 | | | | | | 7 | | | | | | | |
| A4≥ A3? | 4 | | | | | | | | | | | | 10≥7? Да |
| *k*=*k*+... | | 3 | | | | | | | | | | | |
| В ячейку *k* добавляется остаток от деления A[4] на A[3] (10 на 7). Он равен 3. | | | | | | | | | | | | | |
| Обмен A4 и A3 | | | | | | 10 | 7 | | | | | | |
| A5≥ A3? | 5 | | | | | | | | | | | | 7≥10? Нет |
| A6≥ A3? | 6 | | | | | | | | | | | | 9≥10? Нет |
| A7≥ A3? | 7 | | | | | | | | | | | | 11≥10? Да |
| *k*=*k*+... | | 4 | | | | | | | | | | | |
| В ячейку *k* добавляется остаток от деления A[7] на A[3] (11 на 10). Он равен 1. | | | | | | | | | | | | | |
| Обмен A7 и A3 | | | | | | 11 | | | | 10 | | | |
| A8≥ A3? | 8 | | | | | | | | | | | | 7≥11? Нет |
| A9≥ A3? | 9 | | | | | | | | | | | | 8≥11? Нет |

После окончания цикла в переменной *k* лежит число 4.

Ответ: 4.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает два числа: $a$ и $b$. Укажите **наименьшее** из таких чисел $x$, при вводе которого алгоритм печатает сначала 3, а потом 4.

**Бейсик**

```
DIM X, A, B AS INTEGER
INPUT X
A = 0 : B = 0
WHILE X > 0
  IF X MOD 2 > 0 THEN
    A = A + 1
  ELSE
    B = B + 1
  END IF
  X = X \ 2
WEND
PRINT A
PRINT B
```

**Паскаль**

```
var x, a, b: integer;
begin
 readln(x);
 a := 0; b := 0;
 while x > 0 do begin
   if x mod 2 > 0 then
     a := a + 1
   else
     b := b + 1;
   x := x div 2
 end;
 writeln(a); write(b)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, a, b;
  cin >> x;
  a = 0; b = 0;
  while (x > 0) {
    if(x % 2 > 0)
      a = a + 1;
    else
      b = b + 1;
    x = x / 2;
  }
  cout << a << endl << b;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, a, b
  ввод x
  a := 0
  b := 0
  нц пока x > 0
    если mod(x,2) > 0 то
      a := a + 1
    иначе
      b := b + 1
    все
    x := div(x,2)
  кц
  вывод a, нс, b
кон
```

**Python**

```
x = int(input())
a = 0; b = 0
while x > 0:
    if x % 2 > 0:
        a = a + 1
    else:
        b = b + 1
    x = x // 2
print(a)
print(b)
```

Ответ _____.

Р е ш е н и е. Анализируем приведённую программу. В ней видим цикл, в котором, пока переменная $x$ больше нуля, эта переменная делится нацело на 2, и при этом анализируются остатки от деления на 2. Узнаём в этом цикле один из алгоритмов, который требуется знать наизусть при сдаче ЕГЭ по информатике — алгоритм перевода в двоичную систему счисления (пока число не станет равно нулю, делим его нацело на 2 и собираем остатки от деления на 2). При этом, если остаток от деления на 2 оказывается больше нуля (то есть, равен 1), то к значению переменной $a$ добавляется 1. Иначе (то есть, для нуля) увеличивается на 1 значение переменной $b$. То есть, переменная $a$ подсчитывает количество единиц в двоичном представлении числа $x$, а переменная $b$ — количество нулей. Значит, нам нужно по-

добрать такое наименьшее двоичное число, в котором 3 единицы и 4 нуля. Соответственно, число — семиразрядное. Чтобы оно было самым маленьким, его первым разрядом нужно поставить одну единицу (иначе этот разряд не будет значимым), затем написать нули, и в конце — оставшиеся единицы.

Получаем: $1000011_2$. Переведём это число в десятичную систему счисления: $1000011_2 = 2^6 + 2^1 + 2^0 = 64 + 2 + 1 = 67$.

Ответ: 67.

**21** Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B, T, M, R AS INTEGER
A = -30: B = 30
M = A:   R = F(A)
FOR T = A TO B
    IF F(T) < R THEN
       M = T
       R = F(T)
    END IF
NEXT T
PRINT M + R

FUNCTION F (x)
  F = (x*x - 25)*(x*x - 25) + 8
END FUNCTION
```

**Паскаль**

```
var a,b,t,M,R : integer;
function F(x:integer):integer;
begin
    F := (x*x-25)*(x*x-25) + 8
end;
BEGIN
 a := -30;  b := 30;
 M := a;    R := F(a);
 for t := a to b do
   if F(t) < R then
   begin
     M := t;
     R := F(t)
   end;
 write(M + R)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  return (x*x-25)*(x*x-25) + 8;
}
int main() {
  long a, b, t, M, R;
  a = -30;  b = 30;
  M = a;    R = F(a);
  for (t = a; t <= b; t++)
    if ( F(t) < R ) {
      M = t;       R = F(t);
    }
  cout << M + R << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел a, b, t, M, R
  a := -30; b := 30
  M := a;   R := F(a)
  нц для t от a до b
    если F(t) < R то
      M := t;
      R := F(t)
    все
  кц
  вывод M + R
кон

алг цел F(цел x)
нач
  знач := (x*x-25)*(x*x-25) + 8
кон
```

**Python**

```
def F(x):
    return (x*x - 25)*(x*x - 25) + 8
a = -30; b = 30
M = a;   R = F(a)
for t in range(a,b+1):
    if (F(t) < R):
        M = t; R = F(t)
print(M + R)
```

Ответ _____.

Р е ш е н и е. Анализируем программу. Видим, что в ней перебираются целочисленные значения переменной t от -30 до 30. Каждое значение функции в точке F(t) сравнивается с текущим значением переменной R. Если значение F(t) оказывается меньше значения переменной R, значение переменной R меняется на F(t), а в переменную M запоминается текущее значение переменной t. Понимаем, что эта программа ищет минимум и значение минимума функции F(t) на отрезке [-30,30] среди целочисленных значений аргумента.

Анализируем функцию. Понимаем, что она сводится к виду: $F(x) = (x^2 - 5^2)^2 + 8$. Понимаем, что минимум функции равен 8. И что это значение происходит при $(x^2 - 5^2)^2 = 0$. Отсюда находим значения переменной $x$, при которых это выполняется: $x^2 - 5^2 = 0 \Rightarrow x^2 = 5^2 \Rightarrow |x| = 5 \Rightarrow x = \pm 5$.

Среди двух значений нам нужно выбрать то, которое окажется в переменной M. Значения аргумента функции перебираются последовательно от -30 до 30. Соответственно, первым из чисел -5 и 5 будет найдено число -5. Так как при поиске минимума используется строгое неравенство, при значении аргумента, равном 5, условие F(t) < R не выполнится (потому что в этот момент F(t) будет равно R). Поэтому переменная M не изменится и останется равна -5 до конца работы цикла.

На экран выводится результат выражения (M+R). Подставим в него значение -5 и 8.

О т в е т: 3.

**Заметим**, что если бы при поиске минимума использовалось нестрогое неравенство (F(t)<=R), то для t=5 условие бы тоже выполнилось, и в переменной M оказалось бы число 5.

**22** Исполнитель Увеличитель преобразует число на экране.

У исполнителя есть три команды, которым присвоены номера:

**1. Прибавить 1**

**2. Прибавить 2**

**3. Умножить на 3**

Первая команда увеличивает число на экране на 1, вторая увеличивает его на 2, третья умножает его на 3.

Программа для исполнителя Увеличитель — это последовательность команд.

Сколько существует программ, для которых при исходном числе **2** результатом является число **20** и при этом траектория вычислений содержит число **11** и число **16**?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 123 при исходном числе 7 траектория будет состоять из чисел 8, 10, 30.

Ответ _____.

Р е ш е н и е. Так как в задаче требуется, чтобы траектория вычислений содержала числа 11 и 16, вся задача разбивается на три отдельных этапа:

этап 1: сколько способов получить из числа 2 число 11,

этап 2: сколько способов получить из числа 11 число 16,

этап 3: сколько способов получить из числа 16 число 20.

После вычислений нужно будет перемножить все эти три числа.

Для выполнения каждого этапа будем последовательно вычислять количество способов получить из начального числа текущее число. Так, для первого этапа вычислим последовательно количество способов получить из начального числа (2) чисел от 3 до 11.

Для этого построим рекуррентную формулу. Обозначим как $F(n)$ количество способов получить из числа 2 число $n$. Так как, по условию, доступными командами являются «+1», «+2» и «×3», то число $n$ можно получить прибавлением единицы к числу $(n - 1)$, прибавлением двойки к числу $(n - 2)$, и утроением числа $n/3$ (если $n$ кратно трём). То есть, количество способов получить число $n$ равно сумме количества способов получить его «предшественников» — чисел $(n - 1)$, $(n - 2)$ и $n/3$ (если $n$ кратно трём). Запишем это в виде формулы:

$F(n) = F(n - 1) + F(n - 2) + F(n/3)$. Третье слагаемое не учитываем, если $n$ не кратно трём.

Остаётся только понять, что $F(n) = 0$ при $n < 2$ (эти числа нельзя получить операциями «+1», «+2» и «×3».

И что $F(2) = 1$. То есть, существует один способ получить из 2 само число 2 — ничего не делая. Будем последовательно вычислять значения $F(n)$ от 3 до 11.

| $n$ | $F(n)$ | Комментарий |
|---|---|---|
| 2 | 1 | Как мы уже выяснили, начальное число получаем одним способом |
| 3 | 1 | $= F(3-1) + F(3-2) + F(3/3) = F(2) + F(1) + F(1) = 1 + 0 + 0 = 1$ |
| 4 | 2 | $= F(4-1) + F(4-2) = F(3) + F(2) = 1 + 1 = 2$ |
| 5 | 3 | $= F(5-1) + F(5-2) = F(4) + F(3) = 2 + 1 = 3$ |
| 6 | 6 | $= F(6-1) + F(6-2) + F(6/3) = F(5) + F(4) + F(2) = 3 + 2 + 1 = 6$ |
| 7 | 9 | $= F(7-1) + F(7-2) = F(6) + F(5) = 6 + 3 = 9$ |
| 8 | 15 | $= F(8-1) + F(8-2) = F(7) + F(6) = 9 + 6 = 15$ |
| 9 | 25 | $= F(9-1) + F(9-2) + F(9/3) = F(8) + F(7) + F(3) = 15 + 9 + 1 = 25$ |
| 10 | 40 | $= F(10-1) + F(10-2) = F(9) + F(8) = 25 + 15 = 40$ |
| 11 | 65 | $= F(11-1) + F(11-2) = F(10) + F(9) = 40 + 25 = 65$ |

Те же действия можно получить, записав таблицу горизонтально и осуществляя вычисления в уме. Это получается существенно быстрее, но с большей вероятностью ошибки.

| $n$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| $F(n)$ | 1 | 1 | 2 | 3 | 6 | 9 | 15 | 25 | 40 | 65 |

Теперь выполним такие же действия для чисел от 11 до 16.

| $n$ | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|
| $F(n)$ | 1 | 1 | 2 | 3 | 5 | 8 |

Заметим, что на втором этапе нельзя пользоваться значениями $F(n)$, полученными на первом этапе. Потому что на первом этапе мы подсчитывали в $F(n)$ количество способов получить из 2 число $n$. А на втором этапе — число способов получить из 11 число $n$. Эти значения не связаны.

Теперь выполним действия третьего этапа — для чисел от 16 до 20.

| $n$ | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|
| $F(n)$ | 1 | 1 | 2 | 3 | 5 |

В трёх этапах мы получили числа: 65, 8 и 5. Ответ получаем перемножением этих чисел:
$65 \cdot 8 \cdot 5 = 65 \cdot 40 = 2600$.

Ответ: 2600.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ... $x_7$, $x_8$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \wedge x_2) \vee \neg(x_3 \wedge x_4) = 1$
$(x_3 \wedge x_4) \vee \neg(x_5 \wedge x_6) = 1$
$(x_5 \wedge x_6) \vee \neg(x_7 \wedge x_8) = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ... $x_7$, $x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ _____.

Р е ш е н и е. Проанализируем, нельзя ли сделать в задаче замену переменных. То есть, не соответствует ли она условию «каждая пара переменных встречается в задаче в одном единственном выражении относительно друг друга и не пересекается с другими переменными». В данной задаче это так. Действительно, выражение $(x_1 \wedge x_2)$ встречается только один раз, выражение $(x_3 \wedge x_4)$ встречается только в таком виде, и переменные $x_3$ и $x_4$ больше ни в каком другом виде не встречаются. Так же с остальными парами.

Заменим: $y_1 = (x_1 \wedge x_2)$, $y_2 = (x_3 \wedge x_4)$, $y_3 = (x_5 \wedge x_6)$, $y_4 = (x_7 \wedge x_8)$.
Получим систему:
$y_1 \vee \neg y_2 = 1$
$y_2 \vee \neg y_3 = 1$
$y_3 \vee \neg y_4 = 1$

Теперь будем рассматривать по очереди переменные, начиная с $y_1$.

Для удобства начнём с такого значения переменной $y_1$, при котором сразу можно будет сделать какой-нибудь вывод из первого уравнения про $y_2$.

— Пусть $y_1 = 0$. Тогда, чтобы результат $y_1 \vee \neg y_2 = 1$ был верным, нужно чтобы $\neg y_2 = 1$. Отсюда $y_2 = 0$. Если $y_2 = 0$, то из второго уравнения ($y_2 \vee \neg y_3 = 1$) получаем, что $y_3 = 0$. А отсюда и из третьего уравнения делаем вывод, что $y_4 = 0$. Мы дошли до конца «веточки рассуждений». Выяснили, что одним из решений системы относительно переменных $y$ является $y_1 = 0$, $y_2 = 0$, $y_3 = 0$, $y_4 = 0$. Вспомним, что исходно мы решали задачу относительно переменных $x$. Определим, для скольких случаев в парах для переменных $x$ выполняются соответствующие условия для $y$: $y_1 = (x_1 \wedge x_2) = 0$ — 3 случая (когда $x_1 = 0$ или $x_2 = 0$). Так же для $y_2 = 0$, $y_3 = 0$ и $y_4 = 0$. Перемножаем все эти случаи: $3 \cdot 3 \cdot 3 \cdot 3 = \mathbf{81}$.

— Теперь пусть $y_1 = 1$. Тогда из первого уравнения никакого вывода сделать не можем (оно истинно). Переходим к рассмотрению второго уравнения.

— Пусть $y_2 = 0$. Тогда, аналогично первому уравнению и предыдущим размышлениям из второго уравнения получаем, что $y_3 = 0$, а из третьего уравнения — что $y_4 = 0$. Мы дошли до конца «веточки рассуждений». То есть, одним из решений системы относительно переменных $y$ является $y_1 = 1$, $y_2 = 0$, $y_3 = 0$, $y_4 = 0$. Переменная $y_1 = 1$ в одном случае относительно переменных $x$ (когда $x_1 = 1$ и $x_2 = 1$). Число вариантов в этом случае относительно переменных $x$: $1 \cdot 3 \cdot 3 \cdot 3 = \mathbf{27}$.

— Теперь пусть $y_2 = 1$. Тогда второе уравнение истинно. Переходим к рассмотрению третьего уравнения.

— Пусть $y_3 = 0$. Аналогично предыдущим рассуждениям, делаем вывод, что $y_4 = 0$. То есть, одним из решений системы относительно переменных $y$ является $y_1 = 1$, $y_2 = 1$, $y_3 = 0$, $y_4 = 0$. Число вариантов в этом случае относительно переменных $x$: $1 \cdot 1 \cdot 3 \cdot 3 = \mathbf{9}$.

— Теперь пусть $y_3 = 1$. Тогда третье уравнение истинно, и $y_4$ — любое. То есть, одним из решений системы относительно переменных $y$ является $y_1 = 1$, $y_2 = 1$, $y_3 = 1$, $y_4$ любой. Число вариантов в этом случае относительно переменных $x$: $1 \cdot 1 \cdot 1 \cdot 4 = \mathbf{4}$.

Складываем вместе все варианты решения в рассмотренных веточках рассуждений:
$81 + 27 + 9 + 4 = 121$.

Ответ: 121.

**Замечание.** При решении на бумаге все приведённые слова-рассуждения писать не нужно. Поэтому удобнее оформлять решение в виде дерева рассуждений.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используется БЛАНК ОТВЕТОВ № 2. Сначала записывается номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран сумму нечётных цифр числа. Если в числе нет нечётных цифр, программа должна вывести на экран «NO». Ученик торопился и написал программу неправильно.

**Паскаль**

```
var N, s, d: integer;
begin
 readln(N);
 s := 0;
 while N > 0 do
 begin
   d := N mod 10;
   if d mod 2 <> 0 then
       s := s + 1;
   N := N div 10
 end;
 if s mod 2 = 0 then
   writeln('NO')
 else
   writeln(s)
end.
```

**Бейсик**

```
DIM N AS LONG
INPUT N
s = 0
WHILE N > 0
  d = N MOD 10
  IF d MOD 2 <> 0 THEN
    s = s + 1
  END IF
  N = N \ 10
WEND
IF s MOD 2 = 0 THEN
  PRINT "NO"
ELSE
  PRINT s
END IF
END
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  long int N;
  int d, s;
  cin >> N;
  s = 0;
  while (N > 0) {
    d = N % 10;
    if (d % 2 != 0)
        s = s + 1;
    N = N / 10;
  }
  if (s % 2 == 0)
    cout << "NO" << endl;
  else
    cout << s << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N, s, d
  ввод N
  s := 0
  нц пока N > 0
    d := mod(N,10)
    если mod(d,2) <> 0 то
        s := s + 1
    все
    N := div(N, 10)
  кц
  если mod(s,2) = 0 то
    вывод 'NO'
  иначе
    вывод s
  все
кон
```

**Python**

```
N = int(input())
s = 0
while N > 0:
    d = N % 10
    if d % 2 != 0:
        s = s + 1
    N = N // 10
if s % 2 == 0:
    print("NO")
else:
    print(s)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 328.
2. Приведите такое наибольшее трёхзначное число, в котором есть хотя бы одна нечётная цифра, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки: выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**Решение.**

Выполним трассировку программы для данного числа 328.

Рассмотрим подробную трассировку данной программы на языке Паскаль.

| Оператор | Условие | Переменные | | | Пояснения |
|---|---|---|---|---|---|
| | | N | d | s | |
| readln(N); | | 328 | | | Считываем с клавиатуры значение N |
| s := 0; | | | | 0 | Переменная s получает начальное значение (0) |
| while N > 0 do | 328>0? Да | | | | Проверяем условие цикла. Оно выполняется |
| d := N mod 10; | | | 8 | | Переменная d получает значение младшего разряда переменной N |
| if d mod 2<>0 then | 0<>0? Нет | | | | Переменная d — нечётная? Нет. Пропускаем следующую строку |
| N := N div 10 | | 32 | | | Из переменной N удаляем младший разряд |
| while N > 0 do | 32>0? Да | | | | Проверяем условие цикла. Оно выполняется |
| d := N mod 10; | | | 2 | | Переменная d получает значение младшего разряда переменной N |
| if d mod 2<>0 then | 0<>0? Нет | | | | Переменная d — нечётная? Нет. Пропускаем следующую строку |
| N := N div 10 | | 3 | | | Из переменной N удаляем младший разряд |
| while N > 0 do | 3>0? Да | | | | Проверяем условие цикла. Оно выполняется |
| d := N mod 10; | | | 3 | | Переменная d получает значение младшего разряда переменной N |
| if d mod 2<>0 then | 1<>0? Да | | | | Переменная d — нечётная? Да |
| s := s + 1 | | | | 1 | Значение переменной s увеличивается на 1 |
| N := N div 10 | | 0 | | | Из переменной N удаляем младший разряд |
| while N > 0 do | 0>0? Нет | | | | Проверяем условие цикла. Оно не выполняется |
| if s mod 2 = 0 then | 1=0? Нет | | | | Проверяем условие вывода ответа |
| writeln(s) | | | | | На экране: 1 |

На чистовике пишем в ответ на вопрос 1: **1**.

Несмотря на то, что программа выдала правильный ответ, в процессе трассировки мы должны были заметить, что к значению переменной s прибавляется 1, в то время как требуется найти сумму нечётных цифр. Значит, в данном случае, вместо вычисления суммы нечётных цифр, в переменной s вычисляется их количество. То есть, правильно было бы прибавлять значение переменной d.

Также понимаем, что последнее условие (s mod 2 = 0) не всегда правильно работает. Ответ «NO» нужно вывести в случае, когда в числе нет нечётных цифр. А сумма нечётных цифр не обязательно должна быть нечётной. Значит, эта проверка должна быть какой-то другой. Понимаем, что в числе нет нечётных цифр, если их сумма осталась равна начальному значению — нулю. Получается, мы нашли две ошибки в программе:

| Неверно | Верно |
|---|---|
| s := s + 1; | s := s + d; |
| if s mod 2 = 0 then | if s = 0 then |

Проверим ещё раз получившуюся программу. Для этого выполним трассировку исправленной программы для какого-нибудь другого числа. Например, для числа 245. Убеждаемся, что программа выводит правильный ответ (5). Хорошо бы ещё проверить программу для других случаев. Например, для числа 486. В нём нет нечётных цифр, и программа должна выдать ответ «NO». А также для числа 235. В нём несколько нечётных цифр. Программа должна выдать их сумму — 8. Убеждаемся, что программа верно работает. Записываем приведённую выше таблицу в качестве ответа на вопрос 3 (пока оставляем пустую строку для ответа на вопрос 2).

Теперь найдём ответ на вопрос 2. Из анализа программы при поиске ошибок понимаем, что исходная программа вместо суммы нечётных цифр вычисляет их количество. И выводит числовой ответ в том случае, если найденное количество нечётных цифр стало нечётным. Итак, требуется подобрать наибольшее трёхзначное число, в котором нечётное количество нечётных цифр и при этом сумма нечётных цифр равна их количеству. Подберём такое в каждой категории. Сумма нечётных цифр равна их количеству только в том случае, когда все нечётные цифры — единицы. Чтобы количество нечётных цифр было нечётным, оно для трёхзначного числа должно быть 1 или 3. Например, число 111. Однако, так как нам нужно подобрать наибольшее трёхзначное число, правильнее использовать в нём наибольшие цифры. Так как нечётные цифры должны быть только единицы, будем использовать наибольшие чётные цифры. Это 8. Так как у нас должна быть одна единица (вариант 111 мы отбрасываем, потому что он не наибольший), мы поставим эту единицу в конец числа, а восьмерки — в начало. Получаем: 881.

Подставим 881 в исходную программу и проверим, выдаёт ли она при этом правильный ответ (1). Выполним трассировку (можно сделать подробную трассировку, как при ответе на вопрос 1, или сделать упрощенную трассировку). Важно при этом не принимать желаемое за действительное. То есть, точно выполнять операторы один за другим, не внося в этот процесс свои субъективные ощущения, как оно должно бы работать.

Мы не станем приводить здесь всю таблицу. Надеемся, вы сами это сделаете и убедитесь, что программа выдаст число 1. Запишем в качестве ответа на вопрос 2 число 881.

**Замечание.** Рекомендуем ни в коем случае не записывать в качестве ответа ничего кроме указанного. То есть:

1. 1
2. 881
3.

| Неверно | Верно |
|---|---|
| `s := s + 1;` | `s := s + d;` |
| `if s mod 2 = 0 then` | `if s = 0 then` |

**Руководствуйтесь принципом «всё, что сказано, может быть использовано против вас!». В задаче не требуется каким-то образом обосновывать свой ответ или писать какие-либо комментарии. Наоборот, любые комментарии и пояснения, если они будут ошибочны или даже недостаточно точны, будут восприниматься как ошибка! Ничего не пишите в ответ кроме того, что вас спрашивают!**

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно.

Опишите на одном из языков программирования алгоритм, который находит сумму нечётных элементов, не больших 99, а затем заменяет каждый такой элемент на число, равное найденной сумме. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строчки.

Например, для массива из 9-ти элементов: `427 99 300 52 1234 65 16 7 145`
программа должна вывести числа `427 171 300 52 1234 171 16 171 145`

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

**Паскаль**

```
const N = 40;
var a: array [1..N] of integer;
    i, s : integer;
begin
   for i:=1 to N do
     read(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, S AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, s;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, s
  нц для i от 1 до N
    ввод a[i]
  кц
   ...
кон
```

**Python**

```
# допускается также использовать
# целочисленную переменную s
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

**В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).**

Р е ш е н и е.

Сначала найдём требуемую сумму.

Будем перебирать по очереди элементы массива, от начального до конечного. Каждый элемент будем проверять на нечётность и что он не больше 99. Если условие будет выполняться, будем увеличивать значение суммы искомых элементов. Перед циклом не забудем обнулить значение этой суммы.

Очевидно, что пока мы не переберем все элементы массива, мы не сможем получить требуемое значение суммы нечётных элементов, не больших 99. Значит, нельзя в том же цикле и в тот же момент, когда мы находим требуемые элементы, менять их значение на текущее значение суммы — его значение в этот момент ещё не окончательное.

Значит, после окончания цикла нужно ещё раз в цикле перебрать все элементы массива от первого до последнего, ещё раз проверять каждый элемент на нечётность и на то, что его значение не больше 99. Если условие будет выполняться, нужно будет менять значение этого элемента на значение найденной перед этим суммы.

Чтобы не запутаться, рекомендуется после окончания этого цикла запустить ещё раз цикл по всем элементам массива, и вывести значение каждого элемента на экран.

Запишем эту программу.

**Паскаль**

```
s := 0;
for i := 1 to N do
  if (a[i] mod 2 <> 0) and
     (a[i] <= 99) then
    s := s + a[i];
for i := 1 to N do
  if (a[i] mod 2 <> 0) and
     (a[i] <= 99) then
    a[i] := s;
for i := 1 to N do
  writeln(a[i])
```

**Бейсик**

```
S = 0
FOR I = 1 TO N
  IF A(I) MOD 2 <> 0 AND A(I) <= 99 THEN
    S = S + A(I)
  END IF
NEXT I
FOR I = 1 TO N
  IF A(I) MOD 2 <> 0 AND A(I) <= 99 THEN
    A(I) = S
  END IF
NEXT I
FOR I = 1 TO N
  PRINT A(I)
NEXT I
```

**C++**

```
s = 0;
for (i = 0 ; i < N ; i++)
  if (a[i] % 2 != 0 && a[i] <= 99)
    s += a[i];
for (i = 0 ; i < N ; i++)
  if (a[i] % 2 != 0 && a[i] <= 99)
    a[i] = s;
for (i = 0 ; i < N ; i++)
  cout << a[i] << endl;
```

**Алгоритмический язык**

```
s := 0
нц для i от 1 до N
  если mod(a[i],2)<>0 и a[i] <= 99 то
    s := s + a[i]
  все
кц
нц для i от 1 до N
  если mod(a[i],2)<>0 и a[i] <= 99 то
    a[i] := s
  все
кц
нц для i от 1 до N
  вывод a[i]
кц
```

**Python**

```
s = 0
for i in range (0, n):
    if a[i] % 2 != 0 and a[i] <= 99:
        s = s + a[i]
for i in range (0, n):
    if a[i] % 2 != 0 and a[i] <= 99:
        a[i] = s
for i in range (0, n):
    print(a[i])
```

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в кучу **2** камня или **добавить** в кучу **3** камня или увеличить количество камней в куче **в два раза**. Например, имея кучу из 8 камней, за один ход можно получить кучу из 10, 11 или 16 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 51. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **51 или больше** камней.

В начальный момент в куче было S камней, $1 \leqslant S \leqslant 50$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания:

1. а) При каких значениях числа *S* Петя может выиграть первым ходом? Укажите все такие значения и соответствующие выигрывающие ходы.

б) Укажите все такие значения $S$, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Для каждого из указанных значений опишите выигрышную стратегию Вани.

2. Укажите все значения $S$, при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений $S$ опишите выигрышную стратегию Пети.

3. Укажите такое значение $S$, при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения $S$ опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

Р е ш е н и е.

1а) Так как игра заканчивается при $S \geqslant 51$, найдём самое маленькое значение $S$, при котором можно получить 51. Самым «быстрым» ходом, увеличивающим число камней в куче, является удваивание кучи. Найдём $51/2 = 25{,}5$. Это число нецелое. Значит, самым маленьким значением, при котором можно выиграть одним ходом, является число 26. Очевидно, что для всех чисел, которые $\geqslant 26$ из начального диапазона, можно удвоить число камней в куче и получить число, которое $\geqslant 51$. Запишем ответ:

*1а) $S = 26, \ldots, 50$. Первым ходом Петя должен удвоить количество камней в куче.*

1б) Чтобы Ваня победил первым ходом, любой первый ход Пети должен привести игру в позицию, из которой Ваня может одним ходом выиграть. Все такие позиции мы указали в ответе на вопрос 1а. Подберём позиции, из которых любым ходом игра оказывается в диапазоне от 26 до 50 камней. Очевидно, что это так для числа 25 (25+2=27, 25+3=28, 25×2=50). Также это верно для числа 24 (24+2=26, 24+3=27, 24×2=48). Для меньших чисел это будет уже не так, потому что прибавление к числу, меньшему 24 камней, 2-х камней, не даст хотя бы 26 камней. Запишем ответ:

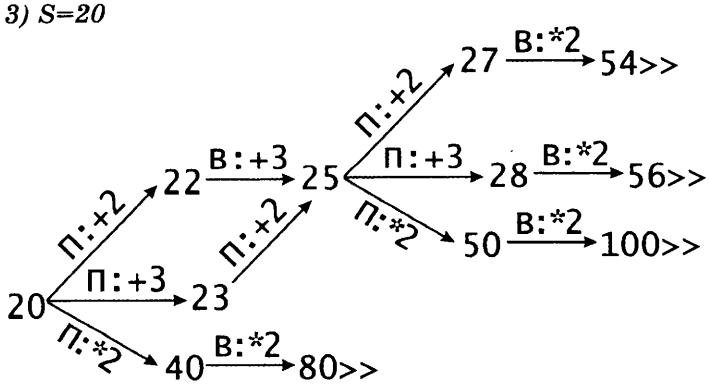
*1б) $S=24$, $S=25$. В обоих случаях, независимо от хода Пети Ваня должен удвоить число камней в куче.*

2) Чтобы Петя выиграл вторым ходом, своим первым ходом он должен перевести игру в позицию проигрыша. Обе возможные позиции мы указали в ответе на вопрос 1б. Найдём позиции, из которых можно получить в куче 24 или 25 камней. Число 24 можно получить из чисел 12, 21 и 22, а число 25 — из чисел 22 и 23. Запишем ответ.

*2) $S=12$, $S=21$, $S=22$, $S=23$. Своим первым ходом Петя должен получить в куче 24 или 25 камней ($12\times2=24$, $21+3=24$, $22+2=24$, $23+2=25$). Независимо от хода Вани, вторым ходом Петя должен удвоить число камней в куче.*

3) Требуемая позиция должна быть такой, чтобы любой первый ход Пети приводил игру в одну из позиций, указанных в ответах на вопросы 1а или 2. Проверим для этого наибольшее из нерассмотренных начальных значений $S$ — 20. Из числа 20 можно получить 22, 23 и 40. Первые два значения — из пункта 2, третье — из пункта 1а. Укажем число 20 в качестве ответа и нарисуем требуемое дерево игры. Для каждой позиции Пети (он проигрывает) будем указывать все возможные ходы. Для каждой позиции Вани — ровно один выигрышный ход. Итак, запишем ответ:

*3) $S=20$*



**Заметим, что в этом пункте обязательно нужно нарисовать полное дерево игры, возможное при выбранной стратегии. Это не то же самое, что описать стратегию игры! Это нельзя делать словами. Здесь нельзя ссылаться на ответы предыдущих пунктов. Дерево возможных партий должно быть нарисовано полностью. Все возможные позиции должны быть обязательно указаны. Не допускается опускать какие-то значения (например, конечные позиции игры). Допускается объединять ветви этого дерева, если они совпадают.**

**27** На вход программе подаётся: в первой строке — число $N$ ($5 < N < 10^4$).
В каждой из последующих $N$ строк — по одному элементу последовательности — целые положительные числа, не превышающее $10^9$.
Напишите эффективную, в том числе и по используемой памяти, программу, которая находит в последовательности такую пару элементов, у которой одновременно выполняются следующие условия:
• максимальная сумма,
• номера элементов различаются не меньше чем на 5,
• сумма должна быть кратна $m$ ($m = 20$),
• элемент пары, который появился в последовательности первым, должен быть больше второго элемента пары.
В качестве ответа необходимо вывести на экран два найденных элемента пары. Гарантируется, что в последовательности всегда найдется такая пара.
Укажите используемую версию языка программирования, например, Free Pascal 2.6.
Программу будем называть эффективной по времени, если при увеличении количества элементов $N$ в $k$ раз и при увеличении параметра $m$ в $k$ раз время работы программы увеличится не более чем в $k$ раз.
Программу будем называть эффективной по памяти, если объём данных, используемый для работы программы, не зависит от количества входных данных $N$ и не превышает 1 килобайта.
За правильную программу, эффективную по памяти и по времени, ставится 4 балла.
За правильную программу, эффективную по времени, но неэффективную по памяти, ставится 3 балла.
За правильную программу, неэффективную по памяти и неэффективную по времени, ставится 2 балла.
Вы можете написать две программы. Итоговый балл за эту задачу вы получите как больший из баллов за эти программы.
*Пример входных данных:*

```
8
13
25
4
86
8
3
7
15
```

*Пример выходных данных:*

```
25 15
```

Решение.

Будем хранить в массиве (буфере) последние 5 элементов последовательности.

В другом массиве для каждого возможного остатка от деления на $m$ будем хранить максимальное значение элемента среди тех, кто хотя бы на 5 номеров «отстоит от текущего».

Будем вычислять для текущего элемента его остаток от деления на $m$ и его «дополнение» (какой нужно иметь остаток от деления на $m$, чтобы сумма элементов делилась на $m$).

Будем проверять, не оказалась ли сумма текущего элемента с его «максимальным дополнением» больше текущей суммы, и, если оказалась, считать эту пару текущим максимумом.

**Затем будем сдвигать буфер влево и дописывать в его конец текущий элемент. Напишем программу на PascalABC.**

```
const m = 20;
var b : array[1..5] of integer;   // буфер
    r : array[0..m-1] of integer; // наибольшие числа для каждого остатка
    a, i, j, m1, m2, p, q, n : integer;
begin
  read(n);
  for i := 1 to 5 do   // считываем первые 5 элементов в буфер
    read(b[i]);
  for i := 0 to m-1 do // обнуляем максимальные числа для каждого остатка
    r[i] := 0;
  m1 := 0;  m2 := 0;   // наибольшие искомые элементы пары
  for i := 6 to n do // перебираем все остальные числа последовательности
  begin
    // обрабатываем первый элемент буфера
    q := b[1] mod m;    // вычисляем его остаток от деления на m
    if b[1] > r[q] then // если b[1] больше текущего максимума
                        // с таким остатком
      r[q] := b[1];     // то обновляем этот максимум
    read(a);            // считываем очередной элемент последовательности
    p := m - a mod m;   // вычисляем «дополнение» к его остатку
    if p = m then       // если оно получилось равно m, то
      p := 0;           // обнуляем это дополнение
    // если левый элемент пары больше правого и
    // если их сумма больше текущего максимума
    if (r[p] > a) and (r[p] + a > m1 + m2) then
    begin
      m1 := r[p];  // обновляем текущую максимальную пару
      m2 := a
    end;
    for j := 1 to 4 do // сдвигаем буфер влево
      b[j] := b[j+1];
    b[5] := a          // кладем в его конец текущий элемент
  end;
  write(m1, ' ', m2)
end.
```

# ВАРИАНТЫ ТРЕНИРОВОЧНЫХ РАБОТ

## ВАРИАНТ 1

### Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько существует натуральных чисел $x$, для которых выполнено неравенство $321_8 < x < DE_{16}$? В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____.

**2** Вася заполнял таблицу истинности функции $z \wedge (y \vee \neg x)$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$:

| | | | $z \wedge (y \vee \neg x)$ |
|---|---|---|---|
| 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Определите, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$.

В ответе напишите буквы $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

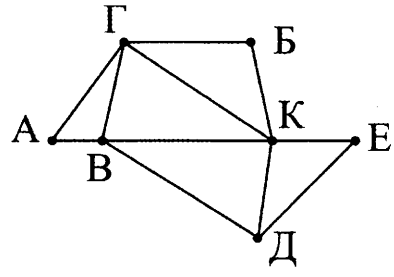
| | | $\neg x \vee y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать $yx$.

Ответ: _____.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|---|---|---|---|---|---|---|---|
| **П1** | | | | 60 | | | 45 |
| **П2** | | | 50 | 20 | 10 | | |
| **П3** | | 50 | | 40 | | 30 | 25 |
| **П4** | 60 | 20 | 40 | | 15 | | 55 |
| **П5** | | 10 | | 15 | | | |
| **П6** | | | 30 | | | | 35 |
| **П7** | 45 | | 25 | 55 | | 35 | |




Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги и пункта Д в пункт К. В ответе запишите целое число — так, как оно указано в таблице.

Ответ: _____.

**4** Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которой также могут встречаться следующие символы.

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находятся шесть файлов:

```
ask.c
kasko.cpp
kraska.doc
nebraska.docx
mask.pic
laska.dpr
```

Ниже представлено восемь масок. Сколько из них таких, которым соответствует ровно три файла из данного каталога?

```
?ask*.???*      *ask*.*c*
*as??.*c*       *ask*.???
*a*a*.*d*       *aska*.?*?
?????*.???      ?s*k?.*
```

Ответ: _____.

**5** Для 5 букв латинского алфавита заданы их двоичные коды. Эти коды представлены в таблице:

| A | B | C | D | E |
|---|---|---|---|---|
| 00 | 100 | 110 | 01 | 101 |

Определите, какой набор букв закодирован двоичной последовательностью 110011000011010101. В ответе запишите последовательность букв без запятых.

Ответ: _____.

**6** Автомат получает на вход два двузначных шестнадцатеричных числа. В этих числах все цифры не превосходят цифру 6 (если в числе есть цифра больше 6, автомат отказывается работать). По этим числам строится новое шестнадцатеричное число по следующим правилам.

1. Вычисляются два шестнадцатеричных числа — сумма старших разрядов полученных чисел и сумма младших разрядов этих чисел.
2. Полученные два шестнадцатеричных числа записываются друг за другом в порядке невозрастания (без разделителей).

*Пример. Исходные числа: 25, 66. Поразрядные суммы: 8, B. Результат: B8.*

Какие из предложенных чисел могут быть результатом работы автомата?

Перечислите в алфавитном порядке буквы, соответствующие этим числам, без пробелов и знаков препинания.

A) 127
B) C6
C) BA
D) E3
E) DA

Ответ: _____.

**7** При работе с электронной таблицей в ячейку C3 записана формула: =$B3+C$2. Какой вид приобретёт формула, после того как ячейку C3 скопируют в ячейку D2?

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 2
s = 8
WHILE s < 500
  s = s + 20
  k = k + 5
WEND
PRINT k
```

**Паскаль**

```
var k, s : integer;
begin
  k := 2;
  s := 8;
  while s < 500 do
  begin
    s := s + 20;
    k := k + 5
  end;
  write(k)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 2;
   s = 8;
   while (s < 500) {
     s = s + 20;
     k = k + 5;
   }
   cout << k << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 2
  s := 8
  нц пока s < 500
    s := s + 20
    k := k + 5
  кц
  вывод k
кон
```

**Python**

```
k = 2
s = 8
while s < 500:
    s = s + 20
    k = k + 5
print(k)
```

Ответ: _____.

**9** Автоматическая фотокамера производит растровые изображения размером 500 на 400 пикселей. При этом объём файла с изображением не может превышать 200 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре?

Ответ: _____.

**10** Все 4-буквенные слова, составленные из букв А, Е, И, О, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. AAAA
2. AAAE
3. AAAИ
4. AAAO
5. AAEA

Запишите слово, стоящее на 248-м месте от начала списка.

Ответ: _____.

**11** Даны рекурсивные алгоритмы F и G. Чему равно значение функции F(6)?

**Бейсик**

```
FUNCTION F(n)
  IF n > 1 THEN
    F = F(n - 1) + G(n - 1)
  ELSE
    F = n
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 2 THEN
    G = G(n - 1) + F(n)
  ELSE
    G = n
  END IF
END FUNCTION
```

**Паскаль**

```
function F(n: integer): integer;
begin
  if n > 1 then
    F := F(n - 1) + G(n - 1)
  else
    F := n
end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n)
  else
    G := n
end;
```

**C++**

```
int F(int n) {
  if (n > 1)
    return F(n-1)+G(n-1);
  else
    return n;
}
int G(int n) {
  if (n > 2)
    return G(n-1)+F(n);
  else
    return n;
}
```

**Алгоритмический язык**

```
алг цел F(цел n)
нач
  если n > 1 то
      знач := F(n - 1) + G(n - 1)
    иначе
      знач := n
  все
кон
алг цел G(цел n)
нач
  если n > 2 то
      знач := G(n - 1) + F(n)
    иначе
      знач := n
  все
кон
```

**Python**

```
def F(n):
    if n > 1:
        return F(n - 1) + G(n - 1)
    else:
        return n
def G(n):
    if n > 2:
        return G(n - 1) + F(n)
    else:
        return n
```

Ответ: _____.

**12** На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес.

В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

| .75 | 5.106 | 21 | 1.24 |
|---|---|---|---|
| **A** | **B** | **C** | **D** |

Ответ: _____.

**13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 10 символов. Из соображений информационной безопасности каждый пароль должен содержать хотя бы 1 десятичную цифру, а также как прописные, так и строчные латинские буквы. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей.

Для хранения сведений о 15 пользователях потребовалось 600 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

Ответ: _____.

**14** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

| вверх | вниз | влево | вправо |
|---|---|---|---|

При выполнении этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у той клетки, где находится РОБОТ:

| сверху свободно | снизу свободно | слева свободно | справа свободно |
|---|---|---|---|

Цикл

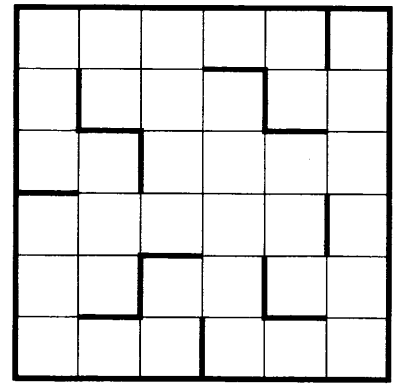
ПОКА **<условие> команда**

Выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

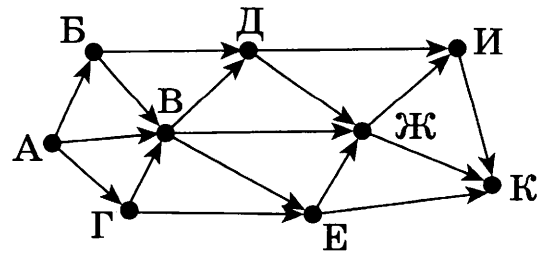
Сколько клеток приведённого лабиринта соответствует требованию, что, выполнив предложенную ниже программу, РОБОТ остановится в той же клетке, с которой он начал движение?

НАЧАЛО
  ПОКА **<сверху свободно> вправо**
  ПОКА **<справа свободно> вниз**
  ПОКА **<снизу свободно> влево**
  ПОКА **<слева свободно> вверх**
КОНЕЦ



Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 32 оканчивается на 4.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц |
|---|---|
| *Яблоки* | 7300 |
| *Яблоки* \| *Сливы* | 14800 |
| *Яблоки & Сливы* | 1400 |

Какое количество страниц будет найдено по запросу: *Сливы* ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

**18** Для какого наибольшего целого числа $A$ формула

$$(x + 2y \neq 60) \vee (A < y) \vee (y < x)$$

тождественно истинна, то есть принимает значение 1 при любых целых неотрицательных $x$ и $y$?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 7; 6; 4; 2; 3; 5; 10; 7; 8 соответственно, т.е. A[0] = 4; A[1] = 7 и т.д.

Определите значение переменной $k$ после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
FOR i = 0 TO 9
  IF A(i) MOD 2 <> 0 THEN
    k = k + A(i)
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
for i := 0 to 9 do
  if A[i] mod 2 <> 0 then
    k := k + A[i];
```

**C++**

```
k = 0;
for (i = 0 ; i <= 9 ; i++)
  if (A[i] % 2 != 0)
    k = k + A[i];
```

**Алгоритмический язык**

```
k := 0
нц для i от 0 до 9
  если mod(A[i],2) <> 0 то
      k := k + A[i]
  все
кц
```

**Python**

```
k = 0
max = A[0]
for i in range(10):
    if A[i] % 2 != 0:
        k = k + A[i]
```

Ответ: _____.

**20** **Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает два числа: $a$ и $b$. Укажите наименьшее из таких чисел $x$, при вводе которого алгоритм печатает сначала 3, а потом 11.**

**Бейсик**

```
DIM X, A, B AS INTEGER
INPUT X
A = 0 : B = 0
WHILE X > 0
  IF X MOD 2 = 0 THEN
    A = A + 1
  END IF
  B = B + X MOD 6
  X = X \ 6
WEND
PRINT A
PRINT B
```

**Паскаль**

```
var x, a, b: integer;
begin
 readln(x);
 a := 0; b := 0;
 while x > 0 do begin
   if x mod 2 = 0 then
     a := a + 1;
   b := b + x mod 6;
   x := x div 6
 end;
 writeln(a); write(b)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, a, b;
  cin >> x;
  a = 0; b = 0;
  while (x > 0) {
    if(x % 2 == 0)
      a = a + 1;
    b = b + x % 6;
    x = x / 6;
  }
  cout << a << endl << b;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, a, b
  ввод x
  a := 0
  b := 0
  нц пока x > 0
    если mod(x,2) = 0 то
      a := a + 1
    все
    b := b + mod(x,6)
    x := div(x,6)
  кц
  вывод a, нс, b
кон
```

**Python**

```
x = int(input())
a = 0; b = 0
while x > 0:
    if x % 2 == 0:
        a = a + 1
    b = b + x % 6
    x = x // 6
print(a)
print(b)
```

**Ответ:** _____.

**21** **Определите, при каком наименьшем значении b в результате выполнения следующего алгоритма будет напечатано число 100 (для Вашего удобства алгоритм представлен на пяти языках):**

**Бейсик**

```
DIM A, B, T, K AS INTEGER
A = 100
INPUT B
K = 0
FOR T = A TO B
  K = K + F(T)
NEXT T
PRINT K

FUNCTION F (x)
  IF x MOD 2 = 0 THEN
    F = 1
  ELSE
    F = 0
  END IF
END FUNCTION
```

**Паскаль**

```
var a,b,t,k : integer;
function F(x:integer):integer;
begin
  if x mod 2 = 0 then
    F := 1
  else
    F := 0
end;
BEGIN
 a := 100;
 readln(b);
 k := 0;
 for t := a to b do
   k := k + F(t);
 write(k)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  if (x % 2 == 0)
    return 1;
  else
    return 0;
}
int main() {
  long a, b, t, k;
  a = 100;
  cin >> b;
  k = 0;
  for (t = a; t <= b; t++)
    k = k + F(t);
  cout << k << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел a, b, t, k
 a := 100
 ввод b
 k := 0
 нц для t от a до b
   k := k + F(t)
 кц
 вывод k
кон

алг цел F(цел x)
нач
   если mod(x,2) = 0 то
     знач := 1
   иначе
     знач := 0
   все
кон
```

**Python**

```
def F(x):
    if x % 2 == 0:
        return 1
    else:
        return 0
a = 100
b = int(input())
k = 0
for t in range(a,b+1):
    k = k + F(t)
print(k)
```

Ответ: _____.

**22** У исполнителя Удвоитель две команды, которым присвоены номера:

**1. прибавь 3,**

**2. умножь на 2.**

Первая из них увеличивает число на экране на 3, вторая — удваивает его.

Программа для Удвоителя — это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 26?

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_9$, $x_{10}$, которые удовлетворяют всем перечисленным ниже условиям?

$(\neg x_1 \equiv \neg x_2) \vee (\neg x_3 \equiv \neg x_4) = 0$

$(\neg x_3 \equiv \neg x_4) \vee (\neg x_5 \equiv \neg x_6) = 0$

$(\neg x_5 \equiv \neg x_6) \vee (\neg x_7 \equiv \neg x_8) = 0$

$(\neg x_7 \equiv \neg x_8) \vee (\neg x_9 \equiv \neg x_{10}) = 0$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_9$, $x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

# Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран сумму нечётных цифр числа. Если в числе нет нечётных цифр, программа должна вывести на экран 0. Программист торопился и написал программу неправильно.

**Паскаль**

```
var N, d, sum: integer;
begin
 readln(N);
 sum := 0;
 while N > 1 do
 begin
   d := N mod 10;
   if d mod 2 <> 0 then
     sum := sum + d;
   N := N div 10
 end;
 writeln(d)
end.
```

**Бейсик**

```
DIM N AS LONG
INPUT N
sum = 0
WHILE N > 1
 d = N MOD 10
 IF d MOD 2 <> 0 THEN
   sum = sum + d
 END IF
 N = N \ 10
WEND
PRINT d
END
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  long int N;
  int d, sum;
  cin >> N;
  sum = 0;
  while (N > 1) {
    d = N % 10;
    if (d % 2 != 0)
      sum = sum + d;
    N = N / 10;
  }
  cout << d << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N, sum, d
  ввод N
  sum := 0
  нц пока N > 1
    d := mod(N,10)
    если mod(d,2) <> 0 то
      sum := sum + d
    все
    N := div(N, 10)
  кц
  вывод d
кон
```

**Python**

```
N = int(input())
sum = 0
while N > 1:
    d = N % 10
    if d % 2 != 0:
        sum = sum + d
    N = N // 10
print(d)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 234.
2. Приведите пример такого числа, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**25** Дан вещественный массив из 40 элементов. Элементы массива могут принимать произвольные значения. Опишите на одном из языков программирования алгоритм, который находит и выводит минимальный положительный элемент массива или сообщение, что такого элемента нет.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of real;
    i, j: integer;
    m : real;
begin
   for i:=1 to N do readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS REAL
DIM I, J AS INTEGER
DIM M AS REAL
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, j;
 double m;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  вещтаб a[1:N]
  цел i, j
  вещ M
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

**Python**

```
# допускается также использовать
# целочисленную переменную j и вещественную переменную m
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в кучу **1** камень или **добавить** в кучу **10** камней. Например, имея кучу из 8 камней, за один ход можно получить кучу из 9 или 18 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 28. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **28 или больше** камней.

В начальный момент в куче было S камней, $1 \leqslant S \leqslant 27$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) При каких значениях числа S Петя может выиграть первым ходом? Укажите все такие значения.

б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения S, при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть за один ход, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для указанных значений S опишите выигрышную стратегию Пети.

3. Укажите такое значение S, при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, но при этом у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

**27** На вход программе подаётся: в первой строке — число N ($5 < N < 10^9$).

В каждой из последующих N строк — по одному элементу последовательности — натуральные числа, не превышающее $10^9$.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая выводит на экран максимальную сумму двух элементов этой последовательности, номера которых различаются не меньше чем на 5.

*Пример входных данных:*

```
8
3
5
4
7
8
3
2
9
```

*Пример выходных данных:*

```
14
```

# ВАРИАНТ 2

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько значащих нулей в двоичной записи числа 188?

Ответ: _____.

**2** Вася заполнял таблицу истинности функции $(y \wedge x) \vee (y \wedge \neg z)$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$:

| | | | $(y \wedge x) \vee (y \wedge \neg z)$ |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Определите, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$.

В ответе напишите буквы $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

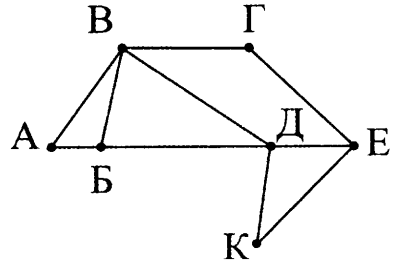
| | | $\neg x \vee y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать $yx$.

Ответ: _____.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|---|---|---|---|---|---|---|---|
| П1 | | 25 | | 13 | 16 | 10 | |
| П2 | 25 | | | | 7 | | |
| П3 | | | | 21 | | 14 | |
| П4 | 13 | | 21 | | | 11 | 5 |
| П5 | 16 | 7 | | | | | 8 |
| П6 | 10 | | 14 | 11 | | | |
| П7 | | | | 5 | 8 | | |




Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги и пункта Г в пункт Е. В ответе запишите целое число — так, как оно указано в таблице.

Ответ: _____.

**4** Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которой также могут встречаться следующие символы.

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

В каталоге находятся пять файлов:

```
boom.pas
bloom.ppt
loom.pas
bottom.pdf
bom.pps
```

Ниже представлено восемь масок. Сколько из них таких, которым соответствует ровно три файла из данного каталога?

```
b*om.???          *oo*m.*p??
?oo*m.p*          *om.*
bo*.??*           ?o*m.p?s
*lo*.p*           b*o*m.??*
```

Ответ: _____.

**5** Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код: А-110, Б-0, В-111, Г-100.

Каким кодовым словом должна кодироваться буква Д? Если таких вариантов несколько, укажите самый короткий.

Ответ: _____.

**6** В некоторой информационной системе информация кодируется двоичными шестиразрядными словами. При передаче данных возможны их искажения, поэтому в конец каждого слова добавляется седьмой (контрольный) разряд таким образом, чтобы сумма разрядов нового слова, считая контрольный, была чётной. Например, к слову 101101 справа будет добавлен 0, а к слову 010110 — 1.

После приёма слова производится его обработка. При этом проверяется сумма его разрядов, включая контрольный. Если она нечётна, это означает, что при передаче этого слова произошёл сбой, и оно автоматически заменяется на зарезервированное слово 0000000. Если она чётна, это означает, что сбоя не было или сбоев было больше одного. В этом случае принятое слово не изменяется.

Приведём фрагмент кодовой таблицы, используемый в данной задаче:

| Буква | Кодовое слово | Примечание |
|---|---|---|
| x | 000000 | сбой |
| A | 100001 | |
| B | 011010 | |
| C | 110010 | |
| D | 110011 | |

Исходное сообщение ABC было передано в виде:

1000010 0110101 1100101

И затем было принято в виде:

1100110 0110101 1100100.

Как будет выглядеть принятое сообщение после обработки? В ответе запишите последовательность букв без пробелов и знаков препинания.

Ответ: _____.

**7** Дан фрагмент электронной таблицы.

| | A | B | C | D |
|---|---|---|---|---|
| **1** | | 3 | 5 | 4 |
| **2** | | =D1+B1 | 1 | 2 |
| **3** | | =C$2+$C3 | 6 | 7 |

Чему станет равным значение ячейки A2, если в неё скопировать формулу из ячейки B3?
*Примечание: знак $ обозначает абсолютную адресацию.*

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
s = 0
FOR k = 3 TO 8
  s = s + k
NEXT k
PRINT s
```

**Паскаль**

```
var k, s : integer;
begin
  s := 0;
  for k := 3 to 8 do
    s := s + k;
  write(s)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   s = 0;
   for(k=3 ; k<=8 ; k++)
     s = s + k;
   cout << s << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  s := 0
  нц для k от 3 до 8
    s := s + k
  кц
  вывод s
кон
```

**Python**

```
s = 0
for k in range(3,9):
    s = s + k
print(s)
```

Ответ: _____.

**9** Производится звукозапись музыкального фрагмента в формате квадро (четырёхканальная запись) с частотой дискретизации 16 кГц и 24-битным разрешением. Результаты записываются в файл, сжатие данных не производится; размер полученного файла 60 Мбайт. Затем производится повторная запись этого же фрагмента в формате стерео (двухканальная запись) с частотой дискретизации 64 кГц и 16-битным разрешением. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

**10** Для кодирования 38-ми различных сообщений используют флажки 3-х видов (красный, зелёный и синий). Сколько флажков нужно использовать для одного сообщения (каждое сообщение кодируется одинаковым числом флажков)?

Ответ: _____.

**11** Алгоритм вычисления значения функции F($n$), где $n$ — натуральное число, задан следующими соотношениями:

F($n$) = F($n$ – 1) + F($n$/2), при $n$ > 1 и $n$ кратно 2

F($n$) = F($n$ – 1) + 1, при $n$ > 1 и $n$ не кратно 2

F(1) = 1

Чему равно значение функции F(8)?

*В ответе запишите только натуральное число.*

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 209.135.205.15

Маска: 255.255.248.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| 0 | 15 | 135 | 200 | 205 | 209 | 248 | 255 |

*Пример.*

*Пусть искомый IP-адрес 192.168.128.0, и дана таблица*

| *A* | *B* | *C* | *D* | *E* | *F* | *G* | *H* |
|---|---|---|---|---|---|---|---|
| *128* | *168* | *255* | *8* | *127* | *0* | *17* | *192* |

*В этом случае правильный ответ будет записан в виде: HBAF*

Ответ: _____.

**13** Для регистрации на сайте онлайн-магазина пользователю предлагается придумать пароль, состоящий из 6 символов. Правилами безопасности сайта разрешается использовать только символы латинского алфавита (строчные и прописные). При этом используется посимвольное кодирование и в памяти сервера для кодирования каждого символа используется минимально возможное и одинаково целое количество бит. А для хранения всего пароля используется минимально возможное целое количество байт. Какое количество информации (в байтах) требуется для хранения паролей 80 пользователей?

Ответ: _____.

**14** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости включает 4 команды-приказа и 4 команды проверки условия.

Команды-приказы:

| **вверх** | **вниз** | **влево** | **вправо** |
|---|---|---|---|

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся.

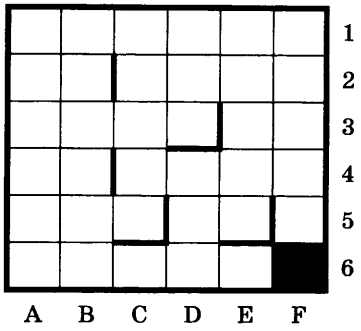
Другие четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

| **сверху свободно** | **снизу свободно** | **слева свободно** | **справа свободно** |
|---|---|---|---|

Цикл
ПОКА <**условие**>
**последовательность команд**
КОНЕЦ ПОКА
выполняется, пока условие истинно.

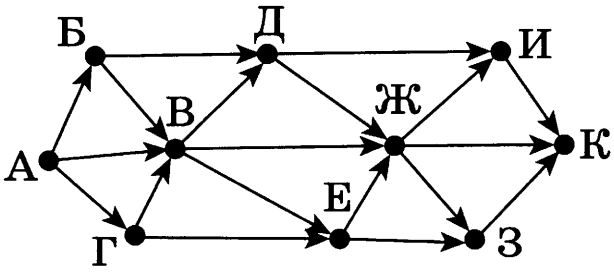
Сколько клеток лабиринта соответствует требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

НАЧАЛО
ПОКА <**справа свободно ИЛИ снизу свободно**>
ПОКА <**справа свободно**>
**вправо**
КОНЕЦ ПОКА
ПОКА <**снизу свободно**>
**вниз**
КОНЕЦ ПОКА
КОНЕЦ ПОКА
КОНЕЦ




Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 23, запись которых в двоичной системе счисления оканчивается на 101.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц |
|---|---|
| *(Клавиатура \| Мышь) & Джойстик* | 11700 |
| *Клавиатура & Джойстик* | 7300 |
| *Мышь & Джойстик* | 5900 |

Какое количество страниц будет найдено по запросу: *Клавиатура & Мышь & Джойстик* ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

**18** Обозначим через m&n поразрядную конъюнкцию неотрицательных целых чисел m и n. Так, например, $14\&5 = 1110_2 \,\&\, 0101_2 = 0100_2 = 4$.

Для какого наименьшего неотрицательного целого числа A формула

(x&45 ≠ 0 ∧ x&A = 0) → x&33 ≠ 0

тождественно истинна (то есть принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 6; 3; 9; 11; 7; 8 соответственно, т.е. A[0] = 4; A[1] = 5 и т.д.

Определите значение переменной *k* после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
FOR i = 0 TO 8
  IF A(i) > A(i+1) THEN
    k = k + 1
    t = A(i)
    A(i) = A(i+1)
    A(i+1) = t
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
for i := 0 to 8 do
  if A[i] > A[i+1] then
  begin
    k := k + 1;
    t := A[i];
    A[i] := A[i+1];
    A[i+1] := t
  end;
```

**C++**

```
k = 0;
for (i = 0 ; i <= 8 ; i++)
  if (A[i] > A[i+1]) {
    k = k + 1;
    t = A[i];
    A[i] = A[i+1];
    A[i+1] = t;
  }
```

**Алгоритмический язык**

```
k := 0
нц для i от 0 до 8
  если A[i] > A[i+1] то
      k := k + 1
      t := A[i]
      A[i] := A[i+1]
      A[i+1] := t
  все
кц
```

**Python**

```
k = 0
for i in range(9):
    if A[i] > A[i+1]:
        k = k + 1
        t = A[i]
        A[i] = A[i+1]
        A[i+1] = t
```

**Ответ:** _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наибольшее из таких чисел *x*, при вводе которых алгоритм печатает сначала 4, а потом 13.

**Бейсик**

```
DIM X, M, L AS INTEGER
INPUT X
M = 0
L = 10
WHILE X > 0
  M = M + X MOD 10
  IF X MOD 10 < L THEN
    L = X MOD 10
  END IF
  X = X \ 10
WEND
PRINT L
PRINT M
```

**Паскаль**

```
var x, M, L: integer;
begin
  readln(x);
  M := 0;
  L := 10;
  while x > 0 do
  begin
    M := M + x mod 10;
    if x mod 10 < L then
      L := x mod 10;
    x := x div 10
  end;
  writeln(L); write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, M, L;
  cin >> x;
  M = 0;
  L = 10;
  while (x > 0) {
    M = M + x % 10;
    if(x % 10 < L)
      L = x % 10;
    x = x / 10;
  }
  cout << L << endl << M;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, M, L
  ввод x
  M := 0
  L := 10
  нц пока x > 0
    M := M + mod(x,10)
    если mod(x,10) < L то
      L := mod(x,10)
    все
    x := div(x,10)
  кц
  вывод L, нс, M
кон
```

**Python**

```
x = int(input())
M = 0; L = 10
while x > 0:
    M = M + x % 10
    if x % 10 < L:
        L = x % 10
    x = x // 10
print(L)
print(M)
```

**Ответ:** _____.

**21** Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B, T, M, R AS INTEGER
A = -20: B = 20
M = A:   R = F(A)
FOR T = A TO B
    IF F(T) <= R THEN
       M = T
       R = F(T)
    END IF
NEXT T
PRINT M + R

FUNCTION F (x)
  F = (x*x - 4)*(x*x - 4) + 8
END FUNCTION
```

**Паскаль**

```
var a,b,t,M,R : integer;
function F(x:integer):integer;
begin
    F := (x*x-4)*(x*x-4) + 8
end;
BEGIN
 a := -20;  b := 20;
 M := a;    R := F(a);
 for t := a to b do
   if F(t) <= R then
   begin
     M := t;
     R := F(t)
   end;
 write(M + R)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  return (x*x - 4)*(x*x - 4)+8;
}
int main() {
  long a, b, t, M, R;
  a = -20;  b = 20;
  M = a;    R = F(a);
  for (t = a; t <= b; t++)
    if ( F(t) <= R ) {
      M = t;
      R = F(t);
    }
  cout << M + R << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел a, b, t, M, R
  a := -20; b := 20
  M := a;   R := F(a)
  нц для t от a до b
    если F(t) <= R то
      M := t
      R := F(t)
    все
  кц
  вывод M + R
кон

алг цел F(цел x)
нач
  знач := (x*x-4)*(x*x-4) + 8
кон
```

**Python**

```
def F(x):
    return (x*x - 4)*(x*x - 4) + 8
a = -20; b = 20
M = a;   R = F(a)
for t in range(a,b+1):
    if (F(t) <= R):
        M = t; R = F(t)
print(M + R)
```

**Ответ:** _____.

**22** У исполнителя Удвоитель две команды, которым присвоены номера:
**1. прибавь 1,**
**2. умножь на 2.**
Первая из них увеличивает число на экране на 1, вторая — удваивает его.
Программа для Удвоителя — это последовательность команд.
Сколько есть программ, которые число 2 преобразуют в число 25?
Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_9$, $x_{10}$, которые удовлетворяют всем перечисленным ниже условиям?
$(\neg x_1 \vee x_2) \vee (\neg x_3 \vee x_4) = 1$
$(\neg x_3 \vee x_4) \wedge (\neg x_5 \vee x_6) = 0$
$(\neg x_5 \vee x_6) \vee (\neg x_7 \vee x_8) = 1$
$(\neg x_7 \vee x_8) \wedge (\neg x_9 \vee x_{10}) = 0$
В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_9$, $x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.
Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры последовательность из $n$ целых чисел ($n = 4$), и выводит на экран сумму чётных чисел среди этой последовательности. Если в последовательности нет чётных чисел, программа выводит «NO». Известно, что вводимые числа не превышают по модулю 1000. Программист торопился и написал программу неправильно.

**Паскаль**

```
const n=4;
var i, a, sum, count: integer;
begin
 sum := 0;
 count := 1;
 for i := 1 to n do
 begin
   read(a);
   if a mod 2 = 0 then
   begin
     sum := sum + a;
     count := count + 1
   end;
 end;
 if sum > 0 then
   writeln(sum)
 else
   writeln('NO')
end.
```

**Бейсик**

```
CONST N=4
DIM I, A, SUM, COUNT AS INTEGER
SUM := 0
COUNT := 1
FOR I = 1 TO N
  INPUT A
  IF A MOD 2 = 0 THEN
    SUM = SUM + A
    COUNT = COUNT + 1
  END IF
NEXT I
IF SUM > 0 THEN
  PRINT SUM
ELSE
  PRINT "NO"
END IF
END
```

**C++**

```
#include <iostream>
using namespace std;
#define n 4
int main() {
  int i, a, sum, count;
  sum = 0;
  count = 1;
  for (i = 0 ; i < n ; i++) {
    cin >> a;
    if(a % 2 == 0) {
      sum = sum + a;
      count = count + 1;
    }
  }
  if (sum > 0)
    cout << sum << endl;
  else
    cout << "NO" << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел n = 4
  цел a, i, sum, count
  sum := 0
  count := 1
  нц для i от 1 до n
    ввод a
    если mod(a,2) = 0
    то
      sum := sum + a
      count := count + 1
    все
  кц
  если sum > 0 то
    вывод sum
  иначе
    вывод 'NO'
  все
кон
```

**Python**

```
N = 4
sum = 0
count = 1
for i in range(n):
    a = int(input())
    if a % 2 == 0:
        sum = sum + a
        count = count + 1
if sum > 0:
    print(sum)
else:
    print("NO")
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел 1 2 3 –4.

2. Приведите пример такой входной последовательности, при которой программа работает верно.

3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно.

Опишите на одном из языков программирования алгоритм, который находит сумму элементов, меньших 300, и при этом не оканчивающихся на 5, а затем заменяет каждый такой элемент на число, равное найденной сумме. Гарантируется, что хотя бы один такой элемент в массиве есть.

В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строчки.

Например, для массива из 9-ти элементов: `427 2 300 5 1234 65 6 7 145`

программа должна вывести числа `427 15 300 5 1234 65 15 15 145`

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

**Паскаль**

```
const N = 40;
var a: array [1..N] of integer;
    i, k, sum : integer;
begin
   for i:=1 to N do
     read(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, K, SUM AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, k, sum;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, k, sum
  нц для i от 1 до N
    ввод a[i]
  кц
   ...
кон
```

**Python**

```
# допускается также
# использовать две
# целочисленные переменные k и sum
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26**

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в кучу **два** или **три** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 17, 18 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 30. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **30 или больше** камней.

В начальный момент в куче было S камней, $1 \leqslant S \leqslant 29$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S, при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S.

б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения S, при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите такое значение S, при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

**27**

На вход программе подаётся последовательность символов, заканчивающаяся символом #. Другие символы # во входной последовательности отсутствуют.

Программа должна вывести на экран латинскую букву, встречающуюся во входной последовательности наибольшее количество раз и число этих раз (во второй строке).

Если таких букв во входной последовательности окажется несколько, программа должна вывести на экран всех их, через пробел, в алфавитном порядке.

Строчные и прописные буквы не различаются.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая должна решать поставленную задачу.

*Пример входных данных:*

```
Day, mice. "Year" - a mistake#
```

*Пример выходных данных:*

```
A
4
```

Другой вариант:

*Пример входных данных:*

```
ABCD ABCE ABCF#
```

*Пример выходных данных:*

```
A B C
3
```

# ВАРИАНТ 3

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Даны 4 целых числа, записанных различных системах счисления: $9F_{16}$, $10110101_2$, $274_8$, $CE_{16}$. Сколько среди них чисел, значение которых лежит между $A5_{16}$ и $CD_{16}$?

Ответ: _____.

**2** Вася заполнял таблицу истинности функции $x \wedge \neg y \wedge ((\neg z) \to w)$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $w$, $x$, $y$, $z$:

| | | | | $x \wedge \neg y \wedge ((\neg z) \to w)$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |

Определите, какому столбцу таблицы соответствует каждая из переменных $w$, $x$, $y$, $z$.

В ответе напишите буквы $w$, $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.*

Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

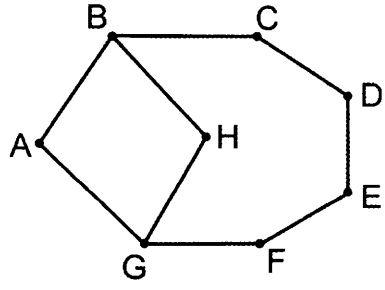
| | | $\neg x \vee y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать $yx$.

Ответ: _____.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о дорогах между населёнными пунктами (звёздочка означает, что дорога между соответствующими городами есть).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 | П8 |
|---|---|---|---|---|---|---|---|---|
| П1 | | | | * | | * | * | |
| П2 | | | * | | * | | | |
| П3 | | * | | | | * | * | |
| П4 | * | | | | | | | * |
| П5 | | * | | | | | | * |
| П6 | * | | * | | | | | |
| П7 | * | | * | | | | | |
| П8 | | | | * | * | | | |



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, номера населённых пунктов **D** и **E** в таблице. В ответе напишите числа в порядке возрастания без разделителей. Например, если бы ответом являлись пункты **П9** и **П8**, в ответе нужно было бы написать 89.

Ответ: _____.

**4** В папке `D:\Pictures\School` находятся следующие файлы и только они:

```
kino.gif
vano.jpg
ono.jpg
domino.gif
no.jpg
```

К этим файлам применили следующую операцию: файлы, имена которых удовлетворяют маске `*?no.*g*` переместили в папку «. .» относительно текущей. Сколько файлов останется в исходной папке?

Ответ: _____.

**5** Для кодирования некоторой последовательности, состоящей из букв A, B, C, D и E, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность.

Вот этот код: A-011, B-000, C-10, D-010, E-001.

Можно ли сократить для одной из букв длину кодового слова так, чтобы код по-прежнему можно было декодировать однозначно?

Коды остальных букв меняться не должны.

Запишите ответ в виде: буква, кодовое слово. Например, если бы ответ был «Для буквы F новый код будет 110», ответ нужно записать как: F110. Если невозможно ничего сократить, запишите в ответе просто цифру 0.

Ответ: _____.

**6** Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются отдельно первая и вторая цифры, а также — вторая и третья цифры.
2. Полученные два числа записываются друг за другом в порядке невозрастания без разделителей.

*Пример.*

*Исходное число: 179.*

*Суммы: 1 + 7 = 8; 7 + 9 = 16.*

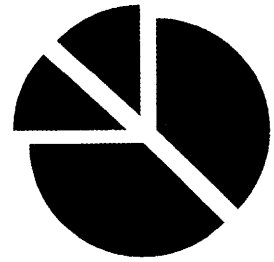
*Результат: 168.*

Укажите наименьшее число, при обработке которого автомат выдаёт результат 126.

Ответ: _____.

**7** Дан фрагмент электронной таблицы:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 3 | | 5 | |
| 2 | =A1/3 | =(A1+C1+1)/3 | =C1-2 | =(B1+C2)/6 |



Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 50
s = 1024
WHILE s > 0
  s = s \ 2
  k = k - 3
WEND
PRINT k
```

**Паскаль**

```
var k, s : integer;
begin
  k := 50;
  s := 1024;
  while s > 0 do
  begin
    s := s div 2;
    k := k - 3
  end;
  write(k)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 50;
   s = 1024;
   while (s > 0) {
     s = s / 2;
     k = k - 3;
   }
   cout << k << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 50
  s := 1024
  нц пока s > 0
    s := div(s,2)
    k := k - 3
  кц
  вывод k
кон
```

**Python**

```
k = 50
s = 1024
while s > 0:
    s = s // 2
    k = k - 3
print(k)
```

**Ответ:** _____.

**9** Производится двухканальная (стерео) звукозапись с частотой дискретизации 16 кГц и 4-битным разрешением. Запись длится одну минуту, её результаты записываются в файл, сжатие данных не производится.

Укажите размер полученного файла (в Мбайт) с точностью 1 Мбайт. В ответе запишите только целое число, единицу измерения писать не нужно.

**Ответ:** _____.

**10** Для кодирования 500 различных сообщений используют 4 последовательных цветовых вспышки.

Лампочки скольких различных цветов должны использоваться при передаче?

**Ответ:** _____.

**11** Определите, сколько звёздочек будет напечатано в результате вызова F(5) приведённой подпрограммы:

**Бейсик**

```
SUB F(n)
  IF n > 1 THEN
    F(n \ 2)
    F(n - 1)
  END IF
  PRINT "*";
END SUB
```

**Паскаль**

```
procedure F(n: integer);
begin
  if n > 1 then
  begin
    F(n div 2);
    F(n - 1)
  end;
  write('*')
end;
```

**C++**

```
void F(int n) {
  if (n > 1) {
    F(n / 2);
    F(n - 1);
  }
  std::cout << "*";
}
```

**Алгоритмический язык**

```
алг F(цел n)
нач
  если n > 1 то
    F(div(n, 2))
    F(n - 1)
  все
  вывод '*'
кон
```

**Python**

```
def F(n):
    if n > 1:
        F(n // 2)
        F(n - 1)
    print("*")
```

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

*Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.*

Для узла с IP-адресом 216.182.73.175 адрес сети равен 216.182.72.0. Чему равно наибольшее возможное количество единиц в разрядах маски?

Ответ: _____.

**13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: A, B, C, D, E, F, G, H, K, L, M, N. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего отведено 12 байт на одного пользователя.

Определите объём памяти (в байтах), необходимый для хранения сведений о 30 пользователях. В ответе запишите только целое число — количество байт.

Ответ: _____.

**14** Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

**А) заменить (v, w).**

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w. Например, выполнение команды **заменить (222, 34)** преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v, то выполнение команды **заменить (v, w)** не меняет эту строку.

**Б) нашлось (v).**

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл
**ПОКА условие**
**последовательность команд**
**КОНЕЦ ПОКА**
выполняется, пока условие истинно.

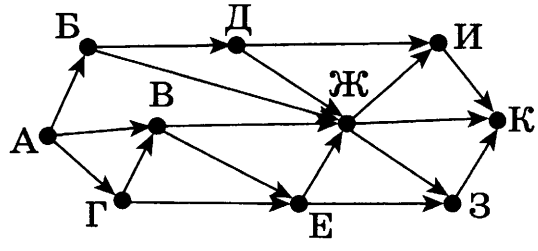
В конструкции
**ЕСЛИ условие**
**ТО команда1**
**ИНАЧЕ команда2**
**КОНЕЦ ЕСЛИ**
выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 80 идущих подряд цифр 7? В ответе запишите полученную строку.

```
НАЧАЛО
ПОКА нашлось (777) ИЛИ нашлось (888)
    ЕСЛИ нашлось (777)
      ТО заменить (777, 8)
      ИНАЧЕ заменить (888, 7)
    КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Значение арифметического выражения:
$9^{200} + 3^{100} - 7$ — записали в системе счисления с основанием 3. Сколько цифр 2 содержится в этой записи?

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| *Колбаса \| Сыр \| Паштет* | 26 |
| *Паштет* | 13 |
| *Сыр* | 8 |
| *Колбаса & Паштет* | 3 |
| *Сыр & Паштет* | 2 |
| *Колбаса & Сыр* | 2 |
| *Колбаса & Сыр & Паштет* | 1 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу *Колбаса* ?

Ответ: _____.

**18** Для какого наименьшего целого числа $A$ формула

$$(xy < A) \vee (x > y) \vee (y \geqslant 9)$$

тождественно истинна, то есть принимает значение 1 при любых целых неотрицательных $x$ и $y$?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 10; 8; 9; 11; 7; 8 соответственно, т.е. A[0]= 4; A[1] = 5 и т.д.

Определите значение переменной $k$ после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
FOR i = 0 TO 9
  IF A(i) >= A(0) THEN
    k = k + 1
    t = A(i)
    A(i) = A(0)
    A(0) = t
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
for i := 0 to 9 do
  if A[i] >= A[0] then
  begin
    k := k + 1;
    t := A[i];
    A[i] := A[0];
    A[0] := t
  end;
```

**C++**

```
k = 0;
for (i = 0 ; i <= 9 ; i++)
  if (A[i] >= A[0]) {
    k = k + 1;
    t = A[i];
    A[i] = A[0];
    A[0] = t;
  }
```

**Алгоритмический язык**

```
k := 0
нц для i от 0 до 9
  если A[i] >= A[0] то
      k := k + 1
      t := A[i]
      A[i] := A[0]
      A[0] := t
  все
кц
```

**Python**

```
k = 0
for i in range(10):
    if A[i] >= A[0]:
        k = k + 1
        t = A[i]
        A[i] = A[0]
        A[0] = t
```

Ответ: _____.

**20** **Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает два числа $L$ и $M$.**

**Укажите наибольшее из таких чисел $x$, при вводе которых алгоритм печатает сначала 3, а потом 7.**

**Бейсик**

```
DIM X, L, M AS INTEGER
INPUT X
L = 0
M = 9
WHILE X > 0
  L = L + 1
  IF M > (X MOD 10) THEN
    M = X MOD 10
  END IF
  X = X \ 10
WEND
PRINT L
PRINT M
```

**Паскаль**

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 9;
  while x > 0 do
   begin
    L := L + 1;
    if M > (x mod 10) then
       M := x mod 10;
    x := x div 10
   end;
  writeln(L);
  write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, L, M;
  cin >> x;
  L = 0;
  M = 9;
  while (x > 0) {
    L = L + 1;
    if(M > x % 10)
      M = x % 10
    x = x / 10;
  }
  cout << L << endl << M;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, L, M
  ввод x
  L := 0
  M := 9
  нц пока x > 0
    L := L + 1
    если M > mod(x,10) то
      M := mod(x,10)
    все
    x := div(x,10)
   кц
   вывод L, нс, M
кон
```

**Python**

```
x = int(input())
L = 0; M = 9
while x > 0:
    L = L + 1
    if M > x % 10:
        M = x % 10
    x = x // 10
print(L)
print(M)
```

**Ответ:** _____.

**21** Определите, при каком наименьшем значении b в результате выполнения следующего алгоритма будет напечатано число 20 (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B AS INTEGER
INPUT B
A = 0
WHILE F(A) < B
  A = A + 1
WEND
PRINT A

FUNCTION F (x)
  IF x = 0
    F = 0
  ELSE
    F = 7 + F(x - 1)
  END IF
END FUNCTION
```

**Паскаль**

```
var a,b : integer;
function F(x:integer):integer;
begin
  if x = 0 then
    F := 0
  else
    F := 7 + F(x - 1)
end;
BEGIN
 readln(b);
 a := 0;
 while F(a) < b do
   a := a + 1;
 write(a)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  if(x == 0)
    return 0;
  else
    return 7 + F(x - 1);
}
int main() {
  int a, b;
  cin >> b;
  a = 0;
  while (F(a) < b)
    a = a + 1;
  cout << a << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел a, b
 ввод b
 a := 0
 нц пока F(a) < b
   a := a + 1
 кц
 вывод a
кон

алг цел F(цел x)
нач
 если x = 0 то
   знач := 0
 иначе
   знач := 7 + F(x - 1)
 все
кон
```

**Python**

```
def F(x):
    if x == 0:
        return 0
    else:
        return 7 + F(x - 1)
b = int(input())
a = 0
while F(a) < b:
    a = a + 1
print(a)
```

Ответ: _____.

**22** У исполнителя Прибавлятель две команды, которым присвоены номера:

**1. прибавь 1,**

**2. прибавь 10.**

Первая из них увеличивает число на экране на 1, вторая — увеличивает его на 10.

Программа для Прибавлятеля — это последовательность команд.

Сколько есть программ, которые число 1 преобразуют в число 24?

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_9$, $x_{10}$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \equiv \neg x_2) \vee (\neg x_3 \equiv x_4) = 0$
$(x_3 \equiv \neg x_4) \vee (\neg x_5 \equiv x_6) = 0$
$(x_5 \equiv \neg x_6) \vee (\neg x_7 \equiv x_8) = 0$
$(x_7 \equiv \neg x_8) \vee (\neg x_9 \equiv x_{10}) = 0$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_9$, $x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры последовательность из $n$ целых чисел ($n = 4$), и выводит на экран максимальное число этой последовательности. Известно, что вводимые числа не превышают по модулю 1000. Программист торопился и написал программу неправильно.

**Паскаль**

```
const n=4;
var i, a, max: integer;
begin
 max := 0;
 for i := 1 to n do
 begin
   read(a);
   if a > max then
     a := max
 end;
 writeln(max)
end.
```

**Бейсик**

```
CONST N=4
DIM I, A, MAX AS INTEGER
MAX := 0
FOR I = 1 TO N
  INPUT A
  IF A > MAX THEN
    A = MAX
  END IF
NEXT I
PRINT MAX
END
```

**C++**

```
#include <iostream>
using namespace std;
const int n = 4;
int main() {
  int i, a, max;
  max = 0;
  for (i = 0 ; i < n ; i++) {
    cin >> a;
    if(a > max)
      a = max;
  }
  cout << max << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел n = 4
  цел a, i, max
  max := 0
  нц для i от 1 до n
    ввод a
    если a > max то
      a := max
    все
  кц
  вывод max
кон
```

**Python**

```
N = 4
max = 0
for i in range(n):
    a = int(input())
    if a > max:
        a = max
print(max)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел 1 2 3 4.

2. Приведите пример такой входной последовательности, при которой программа работает верно.

3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать значения от –10 000 до 10 000. Опишите на одном из языков программирования алгоритм, который находит и выводит наибольшую сумму двух соседних элементов, которая не кратна 3-м. Если таких пар нет, вывести на экран 0.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of integer;
    i, k, sum, m: integer;
begin
   for i:=1 to N do
     readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, K, SUM, M AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, k, sum, m;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, k, sum, m
  нц для i от 1 до N
    ввод a[i]
  кц
   ...
кон
```

**Python**

```
# допускается также использовать три
# целочисленные переменные k, sum и m
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26**

Два игрока, Петя и Ваня, играют в следующую игру.

Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в кучу **один** или **два** камня или увеличить количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 29. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **29 или больше** камней.

В начальный момент в куче было S камней, $1 \leqslant S \leqslant 28$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания.

Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S, при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S.

б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите три значения S, при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите такое значение S, при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

**27**

После единых выпускных экзаменов по информатике в район пришла информация о том, какой ученик какой школы сколько баллов набрал. Эта информация в том же виде была разослана в школы.

Завуч школы № 50 решила наградить двух учащихся, которые лучше всех в школе сдали информатику.

Программа должна вывести на экран фамилии и имена этих учеников.

Если наибольший балл набрало больше двух человек — вывести количество таких учеников.

Если наибольший балл набрал один человек, а следующий балл набрало несколько человек — нужно вывести только фамилию и имя лучшего.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая должна вывести на экран требуемую информацию. Известно, что информатику сдавало больше 5-ти учеников школы № 50.

На вход программе сначала подаётся число учеников, сдававших экзамен. В каждой из следующих N строк находится информация об учениках в формате:

<Фамилия> <Имя> <Номер школы> <Количество баллов>

где <Фамилия> — строка, состоящая не более, чем из 30 символов без пробелов, <Имя> — строка, состоящая не более, чем из 20 символов без пробелов, <Номер школы> — целое число в диапазоне от 1 до 99, <Количество баллов> — целое число диапазоне от 1 до 100. Эти данные записаны через пробел, причём ровно один между каждой парой (то есть, всего по три пробела в каждой строке).

*Пример входной строки:*

```
Иванов Иван 50 87
```

*Пример выходных данных:*

```
Круглов Василий
Тарасова Дарья
```

*Другой вариант выходных данных:*

```
7
```

*Третий вариант выходных данных:*

```
Гусарский Илья
```

# ВАРИАНТ 4

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько чисел находится между $275_8$ и $304_8$?

Ответ: _____.

**2** Дан фрагмент таблицы истинности выражения F. Значения в пустых клетках таблицы неизвестны и могут быть любыми из возможных.

| $x1$ | $x2$ | $x3$ | F |
|---|---|---|---|
| | 1 | 1 | 1 |
| 0 | | 0 | 0 |
| 1 | 1 | | 1 |

Перечислите в порядке возрастания без запятых и пробелов номера логических выражений, которые могут соответствовать F:

1) $\neg x1 \vee x2 \vee \neg x3$
2) $x1 \wedge x2 \wedge x3$
3) $x1 \wedge x2 \wedge \neg x3$
4) $x1 \vee x2 \vee x3$
5) $x1 \wedge x2 \vee x3$

Ответ: _____.

**3** Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | | 3 | 8 | | 4 | |
| B | 3 | | 5 | 9 | 2 | 13 |
| C | 8 | 5 | | 4 | 3 | |
| D | | 9 | 4 | | | 3 |
| E | 4 | 2 | 3 | | | 11 |
| F | | 13 | | 3 | 11 | |

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

Ответ: _____.

**4** Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, какие из указанных имён файлов удовлетворяют маске:

`*la?*.*?o*`

1) `blam.docx`
2) `plan.ozi`
3) `lat.to`
4) `blabla.doc`

В ответе перечислите номера файлов в порядке возрастания без пробелов и знаков препинания.

Ответ: _____.

**5** Для передачи информации по каналу связи с помехами используется 5-битовый код. Сообщение содержит только буквы A, B и C, которые кодируются следующими кодовыми словами:

A-10101, B-10010, C-01110.

Любые два из этих трёх кодовых слов отличаются не менее чем в трёх позициях. Поэтому, если принятое кодовое слово отличается от допустимого не более чем в одной позиции, можно однозначно определить, какая буква передавалась (говорят, что «код исправляет одну ошибку»). Так, получив кодовое слово 10000, можно догадаться, что передавалась буква B (отличие от кодового слова для B только в одной позиции, для остальных кодовых слов отличий больше). Если принятое кодовое слово отличается от всех допустимых кодовых слов более, чем в одной позиции, считается, что произошла ошибка (она обозначается «x»)

Получено сообщение 00101 11010 11110 10111.

Декодируйте это сообщение.

В ответе укажите четыре символа без пробелов и запятых.

Ответ: _____.

**6** Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются отдельно первая и вторая цифры, а также — вторая и третья цифры.
2. Полученные два числа записываются друг за другом в порядке невозрастания без разделителей.

*Пример.*

*Исходное число: 179.*

*Суммы: 1 + 7 = 8; 7 + 9 = 16.*

*Результат: 168.*

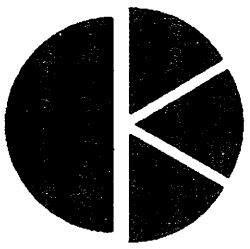
Укажите наименьшее число, при обработке которого автомат выдаёт результат 156.

Ответ: _____.

**7** Дан фрагмент электронной таблицы:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 6 | | | |
| 2 | =D2-B1 | =B1/2 | =A1-B2 | =A1+B1/2 |

Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?



Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 50
s = 0
WHILE s + k > 10
  s = s + 5
  k = k - 15
WEND
PRINT s
```

**Паскаль**

```
var k, s : integer;
begin
  k := 50;
  s := 0;
  while s + k > 10 do
  begin
    s := s + 5;
    k := k - 15
  end;
  write(s)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 50;
   s = 0;
   while (s + k > 10) {
     s = s + 5;
     k = k - 15;
   }
   cout << s << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 50
  s := 0
  нц пока s + k > 10
    s := s + 5
    k := k - 15
  кц
  вывод s
кон
```

**Python**

```
k = 50
s = 0
while s + k > 10:
    s = s + 5
    k = k - 15
print(s)
```

Ответ: _____.

**9** Документ объёмом 4 Мбайта можно передать с одного компьютера на другой двумя способами:

А. Сжать архиватором, передать по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и на сколько, если:

- средняя скорость передачи данных по каналу связи составляет $2^{18}$ бит в секунду;
- объём сжатого архиватором документа равен 50% исходного;
- время, требуемое на сжатие документа, — 20 секунд, на распаковку — 5 секунд?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ: _____.

**10** Эллочка людоедка (в лексиконе которой, как известно, было всего 30 слов) произносит фразу, состоящую из 10 слов. Какое количество информации (количество бит) сообщает Эллочка?

Ответ: _____.

**11** Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(5).

**Бейсик**

```
SUB F(n)
  IF n > 1 THEN
    F(n \ 2)
    PRINT n;
    F(n - 1)
  END IF
END SUB
```

**Паскаль**

```
procedure F(n: integer);
begin
  if n > 1 then
  begin
    F(n div 2);
    write(n);
    F(n - 1)
  end
end;
```

**C++**

```
void F(int n) {
  if (n > 1) {
    F(n / 2);
    std::cout << n;
    F(n - 1);
  }
}
```

**Алгоритмический язык**

```
алг F(цел n)
нач
  если n > 1 то
    F(div(n,2))
    вывод n
    F(n - 1)
  все
кон
```

**Python**

```
def F(n):
    if n > 1:
        F(n // 2)
        print(n)
        F(n - 1)
```

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

*Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.*

Для узла с IP-адресом 205.183.139.102 адрес сети равен 205.183.128.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: _____.

**13** Репетиционный экзамен в некоем регионе сдают 9 потоков по 100 человек в каждом. Каждому из них выделяют специальный код, состоящий из номера потока и номера в потоке. При кодировании этих номеров участников проверяющая система использует минимально возможное количество бит, одинаковое для каждого участника, отдельно номер потока и номер в потоке. При этом для записи кода используется минимально возможное и одинаково целое количество байтов. Каков объём информации (в байтах), записанный устройством после регистрации 80 участников?

Ответ: _____.

**14** Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

**А) заменить (v, w).**

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w. Например, выполнение команды **заменить (222, 34)** преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v, то выполнение команды **заменить (v, w)** не меняет эту строку.

**Б) нашлось (v).**

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл
**ПОКА условие**
**последовательность команд**
**КОНЕЦ ПОКА**
выполняется, пока условие истинно.

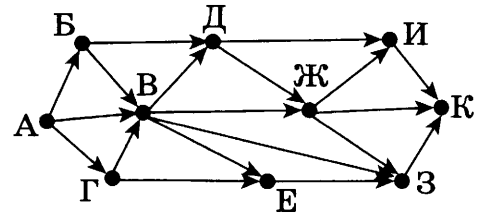
В конструкции
**ЕСЛИ условие**
**ТО команда1**
**ИНАЧЕ команда2**
**КОНЕЦ ЕСЛИ**
выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 70 идущих подряд цифр 5? В ответе запишите полученную строку.

```
НАЧАЛО
ПОКА нашлось (555) ИЛИ нашлось (444)
    ЕСЛИ нашлось (555)
      ТО заменить (555, 4)
      ИНАЧЕ заменить (444, 5)
    КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Какое восьмеричное число находится ровно посередине между числами $10110101_2$ и $C3_{16}$? В ответе напишите только само число в 8-ричной системе счисления. Основание системы счисления писать не нужно.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| *Ангара* | 24 |
| *Лена* | 31 |
| *Енисей* | 19 |
| *Ангара & Лена* | 21 |
| *Ангара & Енисей* | 14 |
| *Лена & Енисей* | 17 |
| *Ангара \| Лена \| Енисей* | 35 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу *Ангара & Лена & Енисей* ?

Ответ: _____.

**18** Укажите наибольшее целое значение $A$, при котором выражение

$$(x < 40) \vee (y < 50) \vee (3x + 2y > A)$$

истинно для любых целых значений $x$ и $y$.

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 7; 7; 10; 8; 9; 11; 7; 6 соответственно, т.е. A[0] = 4; A[1] = 5 и т.д.

Определите значение переменной $k$ после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
FOR i = 0 TO 4
  IF A(2*i) >= A(2*i+1) THEN
    k = k + 1
    t = A(2*i)
    A(2*i) = A(2*i+1)
    A(2*i+1) = t
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
for i := 0 to 4 do
  if A[2*i] >= A[2*i+1] then
  begin
    k := k + 1;
    t := A[2*i];
    A[2*i] := A[2*i+1];
    A[2*i+1] := t
  end;
```

**C++**

```
k = 0;
for (i = 0 ; i <= 4 ; i++)
  if (A[2*i] >= A[2*i+1]) {
    k = k + 1;
    t = A[2*i];
    A[2*i] = A[2*i+1];
    A[2*i+1] = t;
  }
```

**Алгоритмический язык**

```
k := 0
нц для i от 0 до 4
  если A[2*i] >= A[2*i+1] то
      k := k + 1
      t := A[2*i]
      A[2*i] := A[2*i+1]
      A[2*i+1] := t
  все
кц
```

**Python**

```
k = 0
for i in range(5):
    if A[2*i] >= A[2*i+1]:
        k = k + 1
        t = A[2*i]
        A[2*i] = A[2*i+1]
        A[2*i+1] = t
```

Ответ: _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает два числа $L$ и $M$. Укажите наименьшее число $x$, при вводе которого алгоритм печатает сначала 2, а потом 4.

**Бейсик**

```
DIM X, L, M AS INTEGER
INPUT X
L = 0
M = 0
WHILE X > 0
  M = M + 1
  IF X MOD 4 > 1 THEN
    L = L + 1
  END IF
  X = X \ 4
WEND
PRINT L
PRINT M
```

**Паскаль**

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 4 > 1 then
      L := L + 1;
    x := x div 4
  end;
  writeln(L); write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, L, M;
  cin >> x;
  L = 0;
  M = 0;
  while (x > 0) {
    M = M + 1;
    if(x % 4 > 1)
      L = L + 1;
    x = x / 4;
  }
  cout << L << endl << M;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, L, M
  ввод x
  L := 0
  M := 0
  нц пока x > 0
    M := M + 1
    если mod(x,4) > 1 то
      L := L + 1
    все
    x := div(x,4)
  кц
  вывод L, нс, M
кон
```

**Python**

```
x = int(input())
L = 0; M = 0
while x > 0:
    M = M + 1
    if x % 4 > 1:
        L = L + 1
    x = x // 4
print(L)
print(M)
```

**Ответ:** _____.

**21** Определите, при каком наибольшем значении b в результате выполнения следующего алгоритма будет напечатано число 15 (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B AS INTEGER
INPUT B
A = 0
WHILE A < F(B)
  A = A + 1
WEND
PRINT A

FUNCTION F (x)
  IF x < 0
    F = -1
  ELSE
    F = F(x - 4) + 1
  END IF
END FUNCTION
```

**Паскаль**

```
var a,b : integer;
function F(x:integer):integer;
begin
  if x < 0 then
    F := -1
  else
    F := F(x - 4) + 1
end;
BEGIN
 readln(b);
 a := 0;
 while a < F(b) do
   a := a + 1;
 write(a)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  if(x < 0)
    return -1;
  else
    return F(x - 4) + 1;
}
int main() {
  long a, b;
  cin >> b;
  a = 0;
  while (a < F(b))
    a = a + 1;
  cout << a << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел a, b
 ввод b
 a := 0
 нц пока a < F(b)
   a := a + 1
 кц
 вывод a
кон

алг цел F(цел x)
нач
 если x < 0 то
   знач := -1
 иначе
   знач := F(x - 4) + 1
 все
кон
```

**Python**

```
def F(x):
    if x < 0:
        return -1
    else:
        return F(x - 4) + 1
b = int(input())
a = 0
while a < F(b):
    a = a + 1
print(a)
```

Ответ: _____.

**22** Исполнитель Июнь15 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 1**

**2. Умножить на 2**

Первая команда увеличивает число на экране на 1, вторая умножает его на 2.

Программа для исполнителя Июнь15 — это последовательность команд.

Сколько существует программ, для которых при исходном числе **2** результатом является число **34** и при этом траектория вычислений содержит число **12**?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 16, 17.

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_9$, $x_{10}$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \vee \neg x_2) \to (x_3 \vee \neg x_4) = 1$

$(x_3 \vee \neg x_4) \to (x_5 \vee \neg x_6) = 1$

$(x_5 \vee \neg x_6) \to (x_7 \vee \neg x_8) = 1$

$(x_7 \vee \neg x_8) \to (x_9 \vee \neg x_{10}) = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_9$, $x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

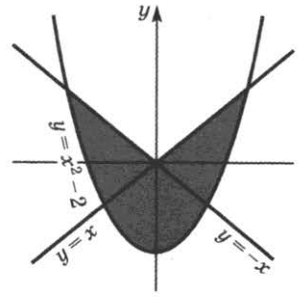
*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

# Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24**

**Требовалось написать программу, при выполнении которой с клавиатуры считываются координаты точки на плоскости (x, y – действительные числа) и определяется принадлежность этой точки заданной закрашенной области (включая границы). Программист торопился и написал программу неправильно.**




**Паскаль**

```
var x,y: real;
begin
 readln(x,y);
 if y <= x then
  if y <= -x then
   if y >= x * x - 2 then
   write('принадлежит')
 else
write('не принадлежит')
end.
```

**Бейсик**

```
INPUT x, y
IF y <= x THEN
 IF y <= -x THEN
  IF y >= x * x - 2 THEN
   PRINT "принадлежит"
    ELSE
 PRINT "не принадлежит"
  ENDIF
 ENDIF
ENDIF
END
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
 float x,y;
 cin >> x >> y;
 if (y <= x)
   if (y <= -x)
     if (y >= x * x - 2)
   cout << "принадлежит" << endl;
 else
 cout << "не принадлежит" << endl;
}
```

**Алгоритмический язык**

```
алг
нач
   вещ x,y
   ввод x,y
   если y <= x то
     если y <= -x то
       если y >= x * x - 2 то
         вывод 'принадлежит'
       иначе
         вывод 'не принадлежит'
       все
     все
   все
кон
```

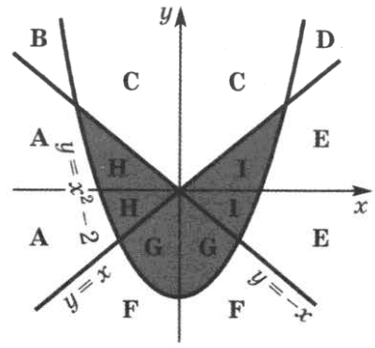
**Python**

```
x = float(input())
y = float(input())
if y <= x:
    if y <= -x:
        if y >= x * x - 2:
            print("принадлежит")
        else:
            print("не принадлежит")
```

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D, E, F, G, H и I).

Точки, лежащие на границах областей, отдельно не рассматривать.




| Область | Условие 1 (y<=x) | Условие 2 (y<= −x) | Условие 3 (y>=x*x−2) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| A | | | | | |
| B | | | | | |
| C | | | | | |
| D | | | | | |
| E | | | | | |
| F | | | | | |
| G | | | | | |
| H | | | | | |
| I | | | | | |

В столбцах условий укажите «да», если условие выполнится, «нет» если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведет себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв». В последнем столбце укажите «да» или «нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев ее неправильной работы. (Это можно сделать несколькими способами, достаточно указать любой способ доработки исходной программы.)

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать значения от −10 000 до 10 000. Опишите на одном из языков программирования алгоритм, который находит и выводит произведение нечётных положительных элементов массива. Если таких элементов нет, вывести на экран 1.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of integer;
    i, j, p: integer;
begin
   for i:=1 to N do
     readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, J, P AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, j, p;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, p
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

**Python**

```
# допускается также использовать
# две целочисленные переменные j и p
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскакль).

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 30. Победителем считается игрок, сделавший последний ход, то есть первым получивший такую позицию, что в кучах всего будет 30 или больше камней.

В начальный момент в первой куче было 4 камня, во второй куче — S камней;

$1 \leqslant S \leqslant 25$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

**Задание 1**

а) Укажите все такие значения числа S, при которых Петя может выиграть за один ход, и соответствующие выигрывающие ходы. Если при некотором значении S Петя может выиграть несколькими способами, достаточно указать один выигрывающий ход.

б) Сколько существует значений S, при которых Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом?

**Задание 2**

Укажите такое значение S, при котором у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

– Петя не может выиграть за один ход;

– Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для указанного значения S опишите выигрышную стратегию Пети.

**Задание 3**

Укажите значение S, при котором одновременно выполняются два условия:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте ходы, в узлах указывайте позиции.

В заданиях 2 и 3 указать одно значение S и объяснить, почему это значение удовлетворяет условию соответствующего задания.

**27**

После единых выпускных экзаменов по информатике в район пришла информация о том, какой ученик какой школы сколько баллов набрал.

В районе считается подозрительной ситуация, когда в школе более двух учащихся набирают одинаковый наибольший балл по школе.

Районный методист решила выяснить таких номера школ.

Программа должна вывести на номера этих школ, в любом порядке.

Если такая школа окажется одна, нужно вывести наибольший балл в этой школе, с указанием того, что это наибольший балл.

Если таких школ не окажется, нужно вывести об этом сообщение.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая должна вывести на экран требуемую информацию. Известно, что информатику сдавало больше 5-ти учеников района. Также известно, что в районе школы с некоторыми номерами не существуют.

На вход программе сначала подаётся число учеников, сдававших экзамен. В каждой из следующих N строк находится информация об учениках в формате:

<Фамилия> <Имя> <Номер школы> <Количество баллов>

где <Фамилия> — строка, состоящая не более, чем из 30 символов без пробелов, <Имя> — строка, состоящая не более, чем из 20 символов без пробелов, <Номер школы> — целое число в диапазоне от 1 до 99, <Количество баллов> — целое число диапазоне от 0 до 100. Эти данные записаны через пробел, причём ровно один между каждой парой (то есть, всего по три пробела в каждой строке).

*Пример входной строки:*

```
Иванов Иван 50 87
```

*Пример выходных данных:*

```
5 50 74 87
```

*Другой вариант выходных данных:*

```
7
Наибольший балл = 74
```

*Третий вариант выходных данных:*

```
Нет таких школ
```

# ВАРИАНТ 5

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Во сколько раз $34_8$ меньше чем $3400_8$? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

**2** Дан фрагмент таблицы истинности выражения F. Значения в пустых клетках таблицы неизвестны и могут быть любыми из возможных.

Перечислите в порядке возрастания без запятых и пробелов номера логических выражений, которые могут соответствовать F:

1) $\neg x1 \wedge x2 \wedge \neg x3$
2) $x1 \vee \neg x2 \vee \neg x3$
3) $\neg x1 \wedge x2 \wedge x3$
4) $\neg x1 \wedge \neg x2 \wedge x3$
5) $x1 \vee \neg x2 \vee x3$

| $x1$ | $x2$ | $x3$ | F |
|---|---|---|---|
| | | | 0 |
| 0 | 1 | 0 | 0 |

Ответ: _____.

**3** Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | | 3 | 8 | | | 16 |
| B | 3 | | 4 | 7 | | |
| C | 8 | 4 | | 2 | 6 | |
| D | | 7 | 2 | | 5 | 4 |
| E | | | 6 | 5 | | 2 |
| F | 16 | | | 4 | 2 | |

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

Ответ: _____.

**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите, у скольких человек, упомянутых в таблице, имеется бабушка, родившаяся раньше, чем за 50 лет до рождения этого человека.

*Таблица 1*

| ID | Фамилия И.О. | Пол | Год |
|---|---|---|---|
| 1518 | Гираб В.А. | М | 1917 |
| 1885 | Жуйко А.И. | М | 1990 |
| 2946 | Калпен Л.В. | М | 1937 |
| 3472 | Арно В.А. | Ж | 2016 |
| 4705 | Иваненко И.Л. | М | 1972 |
| 5624 | Ирита Е.О. | Ж | 1942 |
| 6109 | Кирта У.Т. | Ж | 1995 |
| 7296 | Олинг К.Л. | Ж | 1970 |
| 8434 | Кименко З.Т. | Ж | 1908 |
| 9657 | Вирк Я.М. | Ж | 1965 |
| | ... | | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 1885 | 3472 |
| 6109 | 3472 |
| 9657 | 1885 |
| 4705 | 1885 |
| 2946 | 4705 |
| 5624 | 4705 |
| 8434 | 2946 |
| 1518 | 2946 |
| 2946 | 7296 |
| 5624 | 7296 |
| ... | ... |

Ответ: _____.

**5** Для кодирования некоторой последовательности, состоящей только из четырёх букв А, Б, В, Г, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовали кодовое слово 01; для буквы Б — кодовое слово 1. Какова наименьшая возможная сумма длин кратчайших кодовых слов для букв В и Г?

*Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: _____.

**6** На вход алгоритма подаётся натуральное число N. Алгоритм строит по нему новое число R следующим образом.

1) Строится двоичная запись числа N.

2) К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R.

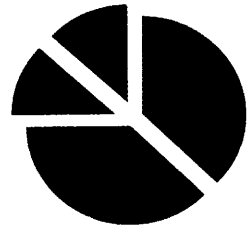
Укажите минимальное число R, которое превышает 60 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Ответ: _____.

**7** Дан фрагмент электронной таблицы:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 3 | | 4 | |
| 2 | =A1/3 | =C1-A2 | =(B1+B2)/2 | =C1-3 |

Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?



Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 70
s = 0
WHILE s + k < 120
  s = s + 15
  k = k - 5
WEND
PRINT s
```

**Паскаль**

```
var k, s : integer;
begin
  k := 70;
  s := 0;
  while s + k < 120 do
  begin
    s := s + 15;
    k := k - 5
  end;
  write(s)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 70;
   s = 0;
   while (s + k < 120) {
     s = s + 15;
     k = k - 5;
   }
   cout << s << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 70
  s := 0
  нц пока s + k < 120
    s := s + 15
    k := k - 5
  кц
  вывод s
кон
```

**Python**

```
k = 70
s = 0
while s + k < 120:
    s = s + 15
    k = k - 5
print(s)
```

Ответ: _____.

**9** Документ объёмом 320 Мбайт можно передать с одного компьютера на другой двумя способами:

А. Сжать архиватором, передать по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и на сколько, если:

- средняя скорость передачи данных по каналу связи составляет $2^{24}$ бит в секунду;
- объём сжатого архиватором документа равен 50% исходного;
- время, требуемое на сжатие документа, — 60 секунд, на распаковку — 30 секунд?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ: _____.

**10** Вася составляет 6-буквенные слова, в которых есть только буквы И, В, А, Н, причём буква А используется в каждом слове ровно 1 раз. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

Ответ: _____.

**11** Даны рекурсивные алгоритмы F и G. Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(15)?

**Бейсик**

```
SUB F(n)
  IF n > 0 THEN
    G(n - 2)
  END IF
END SUB
SUB G(n)
  PRINT n
  IF n > 1 THEN
    F(n - 1)
  END IF
END SUB
```

**Паскаль**

```
procedure G(n: integer); forward;
procedure F(n: integer);
begin
  if n > 0 then G(n - 2)
end;
procedure G(n: integer);
begin
  writeln(n);
  if n > 1 then F(n - 1)
end;
```

**C++**

```
void F(int n);
void G(int n);
void F(int n) {
  if (n > 0)
    G(n - 2);
}
void G(int n) {
  std::cout << n << endl;
  if (n > 1)
    F(n - 1);
}
```

**Алгоритмический язык**

```
алг F(цел n)
нач
  если n > 0 то
    G(n - 2)
  все
кон
алг G(цел n)
нач
  вывод n
  если n > 1 то
    F(n - 1)
  все
кон
```

**Python**

```
def F(n):
    if n > 0:
        G(n - 2)
def G(n):
    print(n)
    if n > 1:
        F(n - 1)
```

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. При этом адрес узла в сети (часть IP-адреса, задающего адрес самого узла в сети) не может состоять из одних нулей или из одних единиц. Сколько компьютеров может быть в сети с маской:

255.255.255.192?

Ответ: _____.

**13** Кабельная сеть проводит голосование среди зрителей о том, какой из пяти фильмов они хотели бы посмотреть вечером. Кабельной сетью пользуются 2000 человек. В голосовании участвовало 1200 человек. Каков объём информации (в байтах), записанный автоматизированной системой голосования?

Ответ: _____.

**14** Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **сместиться на (a, b)**, где $a$, $b$ — целые числа. Эта команда перемещает Чертёжника из точки с координатами $(x, y)$ в точку с координатами $(x + a; y + b)$. Например, если Чертёжник находится в точке с координатами (4, 2), то команда **сместиться на (2, −3)** переместит Чертёжника в точку (6, −1).

Цикл

**ПОВТОРИ число РАЗ**

**последовательность команд**

**КОНЕЦ ПОВТОРИ**

означает, что **последовательность команд** будет выполнена указанное число раз (число должно быть натуральным).

Чертёжнику был дан для исполнения следующий алгоритм (буквами $n$, $a$, $b$ обозначены неизвестные числа, при этом $n > 1$):

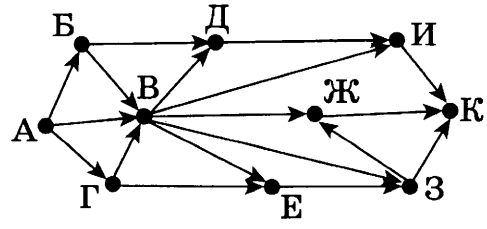
**НАЧАЛО**
  **сместиться на (−5, 8)**
  **ПОВТОРИ n РАЗ**
    **сместиться на (a, b)**
    **сместиться на (14, 18)**
  **КОНЕЦ ПОВТОРИ**
  **сместиться на (−16, −23)**
**КОНЕЦ**

Укажите наименьшее возможное значение числа $n$, для которого найдутся такие значения чисел $a$ и $b$, что после выполнения программы Чертежник возвратится в исходную точку.

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?

Ответ: _____.



**16** В системе счисления с основанием 7 записано трёхзначное число. Известно, что все цифры числа разные. Какое самое большое число удовлетворяет этому условию? Ответ запишите в десятичной системе счисления.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| *Авторучка* \| *Карандаш* \| *Линейка* | 45 |
| *Линейка* | 23 |
| *Карандаш* | 18 |
| *Авторучка* | 15 |
| *Карандаш & Линейка* | 0 |
| *Авторучка & Карандаш* | 8 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу *Авторучка & Линейка* ?

Ответ: _____.

**18** Обозначим через ДЕЛ(n, m) утверждение «натуральное число $n$ делится без остатка на натуральное число $m$».

Для какого наименьшего натурального числа A формула

(¬ДЕЛ(x,12) ∨ ¬ДЕЛ(x,18)) → ¬ДЕЛ(x, A) тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 10; 8; 9; 11; 7; 8 соответственно, т.е. A[0] = 4; A[1] = 5 и т.д.

Определите значение переменной $k$ после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
FOR i = 0 TO 8
  IF A(i) < A(i+1) THEN
    t = A(i)
    A(i) = A(0)
    A(0) = t
    k = k + A(0)
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
for i := 0 to 8 do
  if A[i] < A[i+1] then
  begin
    t := A[i];
    A[i] := A[0];
    A[0] := t;
    k := k + A[0];
  end;
```

**C++**

```
k = 0;
for (i = 0 ; i <= 8 ; i++)
  if (A[i] < A[i+1]) {
    t = A[i];
    A[i] = A[0];
    A[0] = t;
    k = k + A[0];
  }
```

**Алгоритмический язык**

```
k := 0
нц для i от 0 до 8
  если A[i] < A[i+1] то
      t := A[i]
      A[i] := A[0]
      A[0] := t
      k := k + A[0]
  все
кц
```

**Python**

```
k = 0
for i in range(9):
    if A[i] < A[i+1]:
        t = A[i]
        A[i] = A[0]
        A[0] = t
        k = k + A[0]
```

Ответ: _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает число $L$. Известно, что $x > 50$. Укажите наименьшее из таких чисел $x$ (больших 50), при вводе которых алгоритм печатает 6.

**Бейсик**

```
DIM X, L, M AS INTEGER
INPUT X
M = 32
L = X
IF L MOD 7 = 0 THEN
  M = 36
END IF
WHILE M <> 0
  X = L MOD M
  L = M
  M = x
WEND
PRINT L
```

**Паскаль**

```
var x, L, M: integer;
begin
  readln(x);
  M := 32;
  L := x;
  if L mod 7 = 0 then
    M := 36;
  while M  <> 0 do
  begin
    x := L mod M;
    L := M;
    M := x
  end;
  write(L)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, L, M;
  cin >> x;
  M = 32;
  L = x;
  if (L % 7 == 0)
    M = 36;
  while (M != 0) {
    x = L % M;
    L = M;
    M = x;
  }
  cout << L << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, L, M
  ввод x
  M := 32
  L := x
  если mod(L,7) = 0 то
    M := 36
  все
  нц пока M <> 0
    x := mod(L,M)
    L := M
    M := x
  кц
  вывод L
кон
```

**Python**

```
x = int(input())
M = 32; L = x
if L % 7 == 0:
    M = 36
while M != 0:
    x = L % M
    L = M
    M = x
print(L)
```

**Ответ:** _____.

**21** Ниже на пяти языках представлен алгоритм. Напишите в ответе число различных значений входной переменной k, при которых алгоритм выдает тот же ответ, что и при входном значении k = 40. Значение k = 40 также включается в подсчёт различных значений k.

**Бейсик**

```
DIM K, I AS INTEGER
INPUT K
I = 0
WHILE F(I) < K
  I = I + 1
WEND
IF F(I)+F(I-1) < 2*K THEN
  PRINT I
ELSE
  PRINT I-1
END IF

FUNCTION F (n)
    F = n * n + 8
END FUNCTION
```

**Паскаль**

```
var k, i : longint;
function f(n: longint):longint;
begin
  f := n * n + 8
end;
begin
  readln(k);
  i := 0;
  while f(i) < k do
    i := i + 1;
  if f(i)+f(i-1) < 2*k then
    writeln(i)
  else
    writeln(i-1)
end.
```

**C++**

```
#include <iostream>
using namespace std;
long f(long n) {
  return n * n + 8;
}
int main() {
  long k, i;
  cin >> k;
  i = 0;
  while (f(i) < k)
    i++;
  if (f(i) + f(i-1) < 2*k)
    cout << i << endl;
  else
    cout << i - 1 << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел k, i
 ввод k
 i := 0
 нц пока F(i) < k
   i := i + 1
 кц
 если F(i)+F(i-1) < 2 * k то
   вывод i
 иначе
   вывод i - 1
 все
кон

алг цел F(цел n)
нач
 знач := n * n + 8
кон
```

**Python**

```
def f(n):
    return n * n + 8
k = int(input())
i = 0
while f(i) < k:
    i = i + 1
if f(i) + f(i - 1) < 2 * k:
    print(i)
else:
    print(i - 1)
```

Ответ: _____.

**22** Исполнитель Увеличитель преобразует число на экране.

У исполнителя есть три команды, которым присвоены номера:

**1. Прибавить 2**

**2. Умножить на 2**

**3. Умножить на 3**

Первая команда увеличивает число на экране на 2, вторая — умножает его на 2, а третья — умножает его на 3.

Программа для исполнителя Увеличитель — это последовательность команд.

Сколько существует программ, для которых при исходном числе **2** результатом является число **50** и при этом траектория вычислений содержит число **16** и не содержит число **20**?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 123 при исходном числе 7 траектория будет состоять из чисел 8, 16, 48.

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_7$, $x_8$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \equiv \neg x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) = 0$

$(x_2 \equiv \neg x_3) \wedge (x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_4) = 0$

$(x_3 \equiv \neg x_4) \wedge (x_3 \vee x_5) \wedge (\neg x_3 \vee \neg x_5) = 0$

$(x_4 \equiv \neg x_5) \wedge (x_4 \vee x_6) \wedge (\neg x_4 \vee \neg x_6) = 0$

$(x_5 \equiv \neg x_6) \wedge (x_5 \vee x_7) \wedge (\neg x_5 \vee \neg x_7) = 0$

$(x_6 \equiv \neg x_7) \wedge (x_6 \vee x_8) \wedge (\neg x_6 \vee \neg x_8) = 0$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_7$, $x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

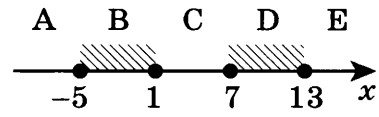
*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

# Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24**

Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (x — действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы). Программист торопился и написал программу неправильно.




**Паскаль**

```
var x: real;
begin
 readln(x);
 if x > 1 then
  if x >= 7 then
   if x <= 13 then
   write('принадлежит')
 else
write('не принадлежит')
end.
```

**Бейсик**

```
INPUT x
IF x > 1 THEN
 IF x >= 7 THEN
  IF x <= 13 THEN
   PRINT "принадлежит"
  ELSE
 PRINT "не принадлежит"
  ENDIF
 ENDIF
END
```

**C++**

```
#include <iostream>
using namespace std;
int main(void) {
 float x;
 cin >> x;
 if (x > 1)
   if (x >= 7)
     if (x <= 13)
   cout << "принадлежит";
 else
 cout << "не принадлежит";
}
```

**Алгоритмический язык**

```
алг
нач
   вещ x
   ввод x
   если x > 1 то
     если x >= 7 то
       если x <= 13 то
         вывод 'принадлежит'
       иначе
         вывод 'не принадлежит'
       все
     все
   все
кон
```

**Python**

```
x = float(input())
if x > 1:
    if x >= 7:
        if x <= 13:
            print("принадлежит")
        else:
            print("не принадлежит")
```

Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки –5, 1, 7 и 13) принадлежат заштрихованным областям (B и D соответственно).

| Область | Условие 1 ($x>1$) | Условие 2 ($x>=7$) | Условие 3 ($x<=13$) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| A | | | | | |
| B | | | | | |
| C | | | | | |
| D | | | | | |
| E | | | | | |

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите «Да» или «Нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать значения от –10 000 до 10 000. Опишите на одном из языков программирования алгоритм, который проверяет, каких значений элементов в массиве больше — положительных (в этом случае вывести «+»), отрицательных (в этом случае вывести «–»), или их поровну (в этом случае вывести «=»).

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of integer;
    i, j, k : integer;
begin
   for i := 1 to N do
     readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, J, K AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, j, k;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

**Python**

```
# допускается также использовать
# две целочисленные переменные j и k
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед ними лежат две кучки камней, в первой из которых 4, а во второй — 3 камня. У каждого игрока неограниченно много камней. Игроки ходят по очереди, первый ход делает Петя. Ход состоит в том, что игрок или **утраивает** число камней в какой-то кучке, или **добавляет 2** камня в какую-то кучку. Игра завершается в тот момент, когда количество камней в одной из кучек становится **не менее 19**. Если в момент завершения игры общее число камней в двух кучках **не менее 33**, то **выиграл Ваня**, в противном случае — Петя. Кто выигрывает при безошибочной игре обоих игроков? Каким должен быть первый ход выигрывающего игрока? Ответ обоснуйте.

**27** На вход программы поступает последовательность из N целых положительных чисел, все числа в последовательности различны. Рассматриваются все пары различных элементов последовательности (элементы пары не обязаны стоять в последовательности рядом, порядок элементов в паре не важен). Необходимо определить количество пар, для которых произведение элементов делится на 6, и при этом номера элементов пары отличаются друг от друга не менее чем на 4.

Описание входных и выходных данных

В первой строке входных данных задаётся количество чисел N ($1 \leqslant N \leqslant 1000$). В каждой из последующих N строк записано одно целое положительное число, не превышающее 10 000.

В качестве результата программа должна напечатать одно число: количество пар, в которых произведение элементов кратно 6, а номера элементов пары отличаются не менее чем на 4.

*Пример входных данных:*

```
7
1
4
3
7
8
3
6
```

*Пример выходных данных для приведённого выше примера входных данных:*

```
5
```

**Пояснение.** Из семи заданных чисел можно составить 6 попарных произведений так, чтобы номера элементов отличались не менее чем на 4: 1 8, 1 3, 1 6, 4 3, 4 6, 3 6, (результаты: 8, 3, 6, 12, 24, 18). Из них на 6 делятся 4 произведения.

Требуется написать эффективную по времени и по памяти программу для решения описанной задачи.

Программа считается эффективной по времени, если при увеличении количества исходных чисел $N$ в $k$ раз время работы программы увеличивается не более чем в $k$ раз.

Программа считается эффективной по памяти, если память, необходимая для хранения всех переменных программы, не превышает 1 Кбайт и не увеличивается с ростом $N$.

Максимальная оценка за правильную (не содержащую синтаксических ошибок и дающую правильный ответ при любых допустимых входных данных) программу, эффективную по времени и по памяти, — 4 балла.

Максимальная оценка за правильную программу, эффективную только по времени — 3 балла.

Максимальная оценка за правильную программу, не удовлетворяющую требованиям эффективности, — 2 балла.

Вы можете сдать одну программу или две программы решения задачи (например, одна из программ может быть менее эффективна). Если Вы сдадите две программы, то каждая из них будет оцениваться независимо от другой, итоговой станет бóльшая из двух оценок.

Перед текстом программы обязательно кратко опишите алгоритм решения.

Укажите использованный язык программирования и его версию.

# ВАРИАНТ 6

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько существует натуральных чисел $x$, для которых выполнено неравенство $93_{16} < x < 236_8$? В ответе укажите только количество чисел, сами числа писать не нужно.

Ответ: _____.

**2** Вася заполнял таблицу истинности функции $(\neg x) \wedge ((\neg y) \vee z)$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$:

| | | | $(\neg x) \wedge ((\neg y) \vee z)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 |

Определите, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$.

В ответе напишите буквы $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

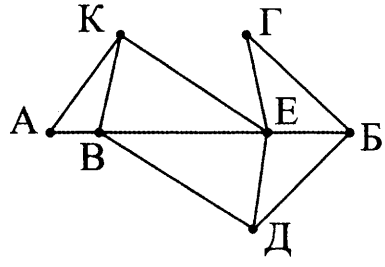
| | | $\neg x \vee y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать $yx$.

Ответ: _____.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 |
|---|---|---|---|---|---|---|---|
| П1 | | | 65 | | | | 60 |
| П2 | | | 30 | 25 | 40 | 45 | |
| П3 | 65 | 30 | | | 55 | 50 | 20 |
| П4 | | 25 | | | | 15 | |
| П5 | | 40 | 55 | | | | 35 |
| П6 | | 45 | 50 | 15 | | | |
| П7 | 60 | | 20 | | 35 | | |




Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги и пункта **Б** в пункт **Д**. В ответе запишите целое число — так, как оно указано в таблице.

Ответ: _____.

**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите на основании приведённых данных ID внука Дени К.Д.

*Таблица 1*

| ID | Фамилия И.О. | Пол |
|---|---|---|
| 1453 | Дени К.Д. | Ж |
| 2185 | Ирта О.Б. | Ж |
| 2605 | Гано И.Е. | Ж |
| 3474 | Ивенко Д.Я. | М |
| 4218 | Бабенко А.Е. | М |
| 5363 | Марченко Е.Е. | М |
| 5771 | Арне А.А. | М |
| 7148 | Камо Е.А. | М |
| 7814 | Исава Г.А. | Ж |
| 8208 | Сорт А.К. | Ж |
| | ... | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 5363 | 4218 |
| 2185 | 4218 |
| 7148 | 5363 |
| 1453 | 5363 |
| 3474 | 1453 |
| 7814 | 1453 |
| 7148 | 2605 |
| 1453 | 2605 |
| 2605 | 8208 |
| 5771 | 7814 |
| ... | ... |

Ответ: _____.

**5** Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв из двух бит, для некоторых из трёх). Эти коды представлены в таблице:

| A | B | C | D | E |
|---|---|---|---|---|
| 110 | 100 | 011 | 010 | 01 |

Определите, какой набор букв закодирован двоичной строкой 01110010011001. В ответе запишите последовательность букв без запятых.

Ответ: _____.

**6** Автомат получает на вход два двузначных шестнадцатеричных числа. В этих числах все цифры не превосходят цифру 7 (если в числе есть цифра больше 7, автомат отказывается работать). По этим числам строится новое шестнадцатеричное число по следующим правилам.

1. Вычисляются два шестнадцатеричных числа — сумма старших разрядов полученных чисел и сумма младших разрядов этих чисел.
2. Полученные два шестнадцатеричных числа записываются друг за другом в порядке неубывания (без разделителей).

*Пример. Исходные числа: 52, 66. Поразрядные суммы: B, 8. Результат: 8B.*

Какие из предложенных чисел могут быть результатом работы автомата?

Перечислите в алфавитном порядке буквы, соответствующие этим числам, без пробелов и знаков препинания.

A) 18
B) 2F
C) 214
D) 7E
E) B7

Ответ: _____.

**7** При работе с электронной таблицей в ячейку `E5` записана формула: `=$C3-B$3`. Какой вид приобретет формула, после того как ячейку `E5` скопируют в ячейку `D3`?

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 1
s = 3
WHILE s < 100
  s = s + 5
  k = k + 2
WEND
PRINT k
```

**Паскаль**

```
var k, s : integer;
begin
  k := 1;
  s := 3;
  while s < 100 do
  begin
    s := s + 5;
    k := k + 2
  end;
  write(k)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 1;
   s = 3;
   while (s < 100) {
     s = s + 5;
     k = k + 2;
   }
   cout << k << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 1
  s := 3
  нц пока s < 100
    s := s + 5
    k := k + 2
  кц
  вывод k
кон
```

**Python**

```
k = 1
s = 3
while s < 100:
    s = s + 5
    k = k + 2
print(k)
```

Ответ: _____.

**9** Автоматическая фотокамера производит растровые изображения размером 600 на 800 пикселей. При этом объём файла с изображением не может превышать 540 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре?

Ответ: _____.

**10** Все 4-буквенные слова, составленные из букв Б, В, Г, Д, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. ББББ
2. БББВ
3. БББГ
4. БББД
5. БВВВ

Запишите слово, стоящее на 244-м месте от начала списка.

Ответ: _____.

**11** Даны рекурсивные алгоритмы F и G. Чему равно значение функции G(6)?

**Бейсик**

```
FUNCTION F(n)
  IF n > 2 THEN
    F = F(n - 1) + G(n - 2)
  ELSE
    F = n
  END IF
END FUNCTION
FUNCTION G(n)
  IF n > 1 THEN
    G = G(n - 1) + F(n)
  ELSE
    G = n + 1
  END IF
END FUNCTION
```

**Паскаль**

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := n
end;
function G(n: integer): integer;
begin
  if n > 1 then
    G := G(n - 1) + F(n)
  else
    G := n + 1
end;
```

**C++**

```
int F(int n) {
  if (n > 2)
    return F(n-1)+G(n-2);
  else
    return n;
}
int G(int n) {
  if (n > 1)
    return G(n-1)+F(n);
  else
    return n + 1;
}
```

**Алгоритмический язык**

```
алг цел F(цел n)
нач
  если n > 2 то
      знач := F(n - 1) + G(n - 2)
    иначе
      знач := n
  все
кон
алг цел G(цел n)
нач
  если n > 1 то
      знач := G(n - 1) + F(n)
    иначе
      знач := n + 1
  все
кон
```

**Python**

```
def F(n):
    if n > 2:
        return F(n - 1) + G(n - 2)
    else:
        return n
def G(n):
    if n > 1:
        return G(n - 1) + F(n)
    else:
        return n + 1
```

**Ответ:** _____.

**12** На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г. Восстановите IP-адрес.

В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

| 7.99 | 31.7 | 2 | 8.21 |
|---|---|---|---|
| **A** | **B** | **C** | **D** |

**Ответ:** _____.

**13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 11 символов. Из соображений информационной безопасности каждый пароль должен содержать как прописные, так и строчные латинские буквы, а также не менее 1 символа из 6-символьного набора: «&», «#», «$», «*», «!», «@». В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей.

Для хранения сведений о 25 пользователях потребовалось 500 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число — количество байт.

Ответ: _____.

**14** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

| вверх | вниз | влево | вправо |
|---|---|---|---|

При выполнении этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →.

Четыре команды проверяют истинность условия отсутствия стены у той клетки, где находится РОБОТ:

| сверху свободно | снизу свободно | слева свободно | справа свободно |
|---|---|---|---|

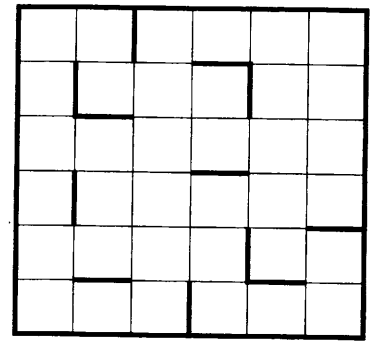
Цикл

**ПОКА <условие> команда**

выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток приведённого лабиринта соответствует требованию, что, выполнив предложенную ниже программу, РОБОТ остановится в той же клетке, с которой он начал движение?

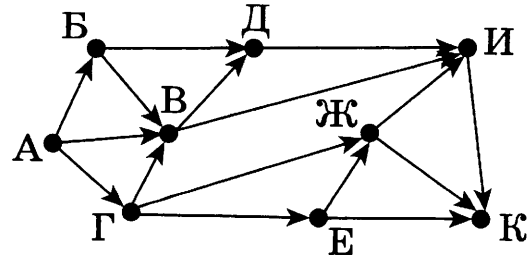


НАЧАЛО
- ПОКА **<слева свободно> вверх**
- ПОКА **<сверху свободно> вправо**
- ПОКА **<справа свободно> вниз**
- ПОКА **<снизу свободно> влево**

КОНЕЦ

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 29 оканчивается на 5.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц |
|---|---|
| *Арбузы* | 9400 |
| *Дыни* | 4700 |
| *Арбузы \| Дыни* | 11900 |

Какое количество страниц будет найдено по запросу: *Арбузы & Дыни* ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

**18** Для какого наименьшего целого числа $A$ формула $(3x + 2y \neq 60) \vee (A > y) \vee (y > x)$ тождественно истинна, то есть принимает значение 1 при любых целых неотрицательных $x$ и $y$?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 6; 5; 4; 7; 10; 6; 9; 11; 7; 8 соответственно, т.е. A[0] = 6; A[1] = 5 и т.д.

Определите значение переменной $k$ после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
min1 = A(0)
FOR i = 1 TO 9
  IF A(i) <= min1 THEN
    k = i
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
min1 := A[0];
for i:=1 to 9 do
  if A[i] <= min1 then
    k := i;
```

**C++**

```
k = 0;
min1 = A[0];
for (i = 1 ; i <= 9 ; i++)
  if(A[i] <= min1)
    k = i;
```

**Алгоритмический язык**

```
k := 0
min1 := A[0]
нц для i от 1 до 9
  если A[i] <= min1 то
    k := i
  все
кц
```

**Python**

```
k = 0
min1 = A[0]
for i in range(1,10):
    if A[i] <= min1:
        k = i
```

Ответ: _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает два числа: $L$ и $M$. Укажите наименьшее из таких чисел $x$, при вводе которых алгоритм печатает сначала 8, а потом 13.

**Бейсик**

```
DIM X, M, L AS INTEGER
INPUT X
M = 0
L = 0
WHILE X > 0
  M = M + X MOD 10
  IF X MOD 10 > L THEN
    L = X MOD 10
  END IF
  X = X \ 10
WEND
PRINT L
PRINT M
```

**Паскаль**

```
var x, M, L: integer;
begin
  readln(x);
  M := 0; L := 0;
  while x > 0 do
  begin
    M := M + x mod 10;
    if x mod 10 > L then
      L := x mod 10;
    x := x div 10
  end;
  writeln(L); write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, M, L;
  cin >> x;
  M = 0;
  L = 0;
  while (x > 0) {
    M = M + x % 10;
    if(x % 10 > L)
      L = x % 10;
    x = x / 10;
  }
  cout << L << endl << M;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, M, L
  ввод x
  M := 0
  L := 0
  нц пока x > 0
    M := M + mod(x,10)
    если mod(x,10) > L то
      L := mod(x,10)
    все
    x := div(x,10)
  кц
  вывод L, нс, M
кон
```

**Python**

```
x = int(input())
M = 0; L = 0
while x > 0:
    M = M + x % 10
    if x % 10 > L:
        L = x % 10
    x = x // 10
print(L)
print(M)
```

**Ответ:** _____.

**21** Определите, при каком наибольшем значении b в результате выполнения следующего алгоритма будет напечатано число 50 (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B, T, K AS INTEGER
A = 20
INPUT B
K = 0
FOR T = A TO B
  K = K + F(T)
NEXT T
PRINT K

FUNCTION F (x)
  IF x MOD 3 = 0 THEN
    F = 0
  ELSE
    F = 1
  END IF
END FUNCTION
```

**Паскаль**

```
var a,b,t,k : integer;
function F(x:integer):integer;
begin
  if x mod 3 = 0 then
    F := 0
  else
    F := 1
end;
BEGIN
 a := 20;
 readln(b);
 k := 0;
 for t := a to b do
   k := k + F(t);
 write(k)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  if (x % 3 == 0)
    return 0;
  else
    return 1;
}
int main() {
  int a, b, t, k;
  a = 20;
  cin >> b;
  k = 0;
  for (t = a; t <= b; t++)
    k = k + F(t);
  cout << k << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел a, b, t, k
 a := 20
 ввод b
 k := 0
 нц для t от a до b
   k := k + F(t)
 кц
 вывод k
кон

алг цел F(цел x)
нач
   если mod(x,3) = 0 то
     знач := 0
   иначе
     знач := 1
   все
кон
```

**Python**

```
def F(x):
    if x % 3 == 0:
        return 0
    else:
        return 1
a = 20
b = int(input())
k = 0
for t in range(a,b+1):
    k = k + F(t)
print(k)
```

Ответ: _____.

**22** У исполнителя Утроитель две команды, которым присвоены номера:

**1. прибавь 1,**

**2. умножь на 3.**

Первая из них увеличивает число на экране на 1, вторая утраивает его.

Программа для Утроителя — это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 28?

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_7$, $y_1$, $y_2$, ..., $y_7$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \to (x_2 \wedge \neg y_1)) \wedge (y_1 \to y_2) = 1$

$(x_2 \to (x_3 \wedge \neg y_2)) \wedge (y_2 \to y_3) = 1$

$(x_3 \to (x_4 \wedge \neg y_3)) \wedge (y_3 \to y_4) = 1$

$(x_4 \to (x_5 \wedge \neg y_4)) \wedge (y_4 \to y_5) = 1$

$(x_5 \to (x_6 \wedge \neg y_5)) \wedge (y_5 \to y_6) = 1$

$(x_6 \to (x_7 \wedge \neg y_6)) \wedge (y_6 \to y_7) = 1$

$x_7 \to \neg y_7 = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_7$, $y_1$, $y_2$, ..., $y_7$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

# Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран наибольшую цифру числа. Программист торопился и написал программу неправильно.

**Паскаль**

```
var N,k: integer;
begin
 readln(N);
 k := 10;
 while N > 0 do
 begin
   if N mod 10 < k then
     k := N mod 10;
   N := N div 10
 end;
 writeln(k)
end.
```

**Бейсик**

```
DIM N AS LONG
INPUT N
k = 10
WHILE N > 0
 IF N MOD 10 < k THEN
   k = N MOD 10
 END IF
 N = N \ 10
WEND
PRINT k
END
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  long int N;
  int k;
  cin >> N;
  k = 10;
  while (N > 0) {
    if (N % 10 < k)
      k = N % 10;
    N = N / 10;
  }
  cout << k << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N, k
  ввод N
  k := 10
  нц пока N > 0
    если mod(N,10) < k то
      k := mod(N,10)
    все
    N := div(N, 10)
  кц
  вывод k
кон
```

**Python**

```
N = int(input())
k = 10
while N > 0:
    if N % 10 < k:
        k = N % 10
    N = N // 10
print(k)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 527.
2. Приведите пример такого числа, при котором программа работает верно.
3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**25** Дан целочисленный массив из 30 элементов. Элементы массива могут принимать произвольные значения. С клавиатуры вводится целое число X. Опишите на одном из языков программирования алгоритм, который находит и выводит наименьший номер элемента массива, равного X, или сообщение, что такого элемента нет. При выводе ответа считать, что элементы массива нумеруются с единицы (учесть это при выводе для языков программирования, нумерующих элементы массива с нуля).

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 30;
var
    a: array [1..N] of integer;
    i, j, x: integer;
begin
   for i:=1 to N do readln(a[i]);
   readln(x);
   ...
end.
```

**Бейсик**

```
N = 30
DIM A(N) AS INTEGER
DIM I, J, X AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
INPUT X
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 30;
int main() {
 long a[N];
 long i, j, x;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 30
  целтаб a[1:N]
  цел i, j, x
  нц для i от 1 до N
    ввод a[i]
  кц
  ввод x
  ...
кон
```

**Python**

```
# допускается также использовать две
# целочисленные переменные j и x
a = []
n = 30
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 31. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **31 или больше** камней.

В начальный момент в куче было S камней, $1 \leqslant S \leqslant 30$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S, при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S.

б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения S, при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите такое значение S, при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

**27**

*Вам предлагаются два задания, связанные с этой задачей: задание А и задание Б. Вы можете решать оба задания А и Б или одно из них по своему выбору.*

*Итоговая оценка выставляется как максимальная из оценок за задания А и Б.*

*Если решение одного из заданий не представлено, то считается, что оценка за это задание составляет 0 баллов.*

*Задание Б является усложненным вариантом задания А, оно содержит дополнительные требования к программе.*

Напишите программу, которая должна вывести на экран минимальное чётное произведение двух элементов последовательности, номера которых различаются не меньше чем на 6.

Если такой пары элементов нет, программа должна вывести ноль.

**А.** Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов.

Перед программой укажите версию языка программирования.

**Обязательно** укажите, что программа является решением **задания А**.

Максимальная оценка за выполнение задания А — 2 балла.

**Б.** Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству элементов последовательности N, т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 Кбайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

Обязательно укажите, что программа является решением задания Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, — 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, — 3 балла.

**Напоминаем!** Не забудьте указать, к какому заданию относится каждая из представленных вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число N — общее количество элементов последовательности ($N \leqslant 10\,000$).

В каждой из следующих N строк задается одно положительное целое число — очередной элемент последовательности.

Известно, что каждое число положительное и не превышает 1000.

*Пример входных данных:*

```
8
1
4
5
7
8
3
2
9
```

*Пример выходных данных:*

```
2
```

# ВАРИАНТ 7

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько значащих нулей в двоичной записи числа 237?

Ответ: _____.

**2** Вася заполнял таблицу истинности функции $(y \wedge \neg z) \vee (\neg x \wedge \neg z)$, но успел заполнить лишь фрагмент из трёх различных ее строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$:

| | | | $(y \wedge \neg z) \vee (\neg x \wedge \neg z)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |

Определите, какому столбцу таблицы соответствует каждая из переменных $x$, $y$, $z$.

В ответе напишите буквы $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

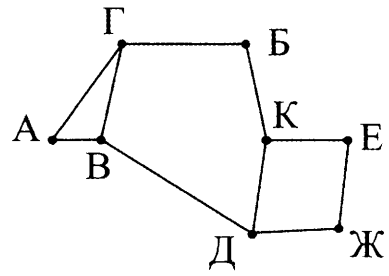
| | | $\neg x \vee y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать $yx$.

Ответ: _____.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 | П8 |
|---|---|---|---|---|---|---|---|---|
| П1 | | | | | | 10 | 14 | 21 |
| П2 | | | 11 | | | | 18 | |
| П3 | | 11 | | 7 | | | | 23 |
| П4 | | | 7 | | 25 | | | |
| П5 | | | | 25 | | | | 12 |
| П6 | 10 | | | | | | 19 | |
| П7 | 14 | 18 | | | | 19 | | |
| П8 | 21 | | 23 | | 12 | | | |




Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги и пункта А в пункт Г. В ответе запишите целое число — так, как оно указано в таблице.

Ответ: _____.

**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите на основании приведённых данных ID дяди Сорт А.К. (Дядя — брат отца или матери, а также муж тети (сестры отца или матери)).

*Таблица 1*

| ID | Фамилия И.О. | Пол |
|---|---|---|
| 1453 | Дени К.Д. | Ж |
| 2185 | Ирта О.Б. | Ж |
| 2605 | Гано И.Е. | Ж |
| 3474 | Ивенко Д.Я. | М |
| 4218 | Бабенко А.Е. | М |
| 5363 | Марченко Е.Е. | М |
| 5771 | Арне А.Е. | Ж |
| 7148 | Камо Е.А. | М |
| 7814 | Исава Г.А. | Ж |
| 8208 | Сорт А.К. | Ж |
| | ... | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 5363 | 4218 |
| 2185 | 4218 |
| 7148 | 5363 |
| 1453 | 5363 |
| 3474 | 1453 |
| 7814 | 1453 |
| 7148 | 2605 |
| 1453 | 2605 |
| 2605 | 8208 |
| 5363 | 5771 |
| ... | ... |

Ответ: _____.

**5** Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г и Д, используется неравномерный двоичный код, позволяющий однозначно декодировать полученную двоичную последовательность. Вот этот код:

А-010, Б-011, В-000, Г-001. Каким кодовым словом должна кодироваться буква Д? Если таких вариантов несколько, укажите самый короткий.

Ответ: _____.

**6** В некоторой информационной системе информация кодируется двоичными шестиразрядными словами. При передаче данных возможны их искажения, поэтому в конец каждого слова добавляется седьмой (контрольный) разряд таким образом, чтобы сумма разрядов нового слова, считая контрольный, была чётной. Например, к слову 101101 справа будет добавлен 0, а к слову 010110 — 1.

После приёма слова производится его обработка. При этом проверяется сумма его разрядов, включая контрольный. Если она нечётна, это означает, что при передаче этого слова произошел сбой, и оно автоматически заменяется на зарезервированное слово 0000000. Если она чётна, это означает, что сбоя не было или сбоев было больше одного. В этом случае принятое слово не изменяется.

Приведём фрагмент кодовой таблицы, используемый в данной задаче:

| Буква | Кодовое слово | Примечание |
|---|---|---|
| x | 000000 | сбой |
| A | 101011 | |
| B | 011001 | |
| C | 110111 | |
| D | 110110 | |

Исходное сообщение ABC было передано в виде:

1010110 0110011 1101111

И затем было принято в виде:

1110110 0110001 1101100.

Как будет выглядеть принятое сообщение после обработки? В ответе запишите последовательность букв без пробелов и знаков препинания.

Ответ: _____.

**7** Дан фрагмент электронной таблицы.

| | A | B | C | D |
|---|---|---|---|---|
| **1** | | 3 | 5 | 4 |
| **2** | 10 | =C$3+B3 | 1 | 2 |
| **3** | 20 | =C$2+$C3 | 6 | 7 |

Чему станет равным значение ячейки A1, если в нее скопировать формулу из ячейки B2?
*Примечание: знак $ обозначает абсолютную адресацию.*
Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**
```
DIM k, s AS INTEGER
s = 1
FOR k = 4 TO 7
  s = s * k
NEXT k
PRINT s
```

**Паскаль**
```
var k, s : integer;
begin
  s := 1;
  for k := 4 to 7 do
    s := s * k;
  write(s)
end.
```

**C++**
```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   s = 1;
   for(k=4 ; k<=7 ; k++)
     s = s * k;
   cout << s << endl;
   return 0;
}
```

**Алгоритмический язык**
```
алг
нач
  цел k, s
  s := 1
  нц для k от 4 до 7
    s := s * k
  кц
  вывод s
кон
```

**Python**
```
s = 1
for k in range(4,8):
    s = s * k
print(s)
```

Ответ: _____.

**9** Производится звукозапись музыкального фрагмента в формате квадро (четырёхканальная запись) с частотой дискретизации 32 кГц и 24-битным разрешением. Результаты записываются в файл, сжатие данных не производится; размер полученного файла 90 Мбайт. Затем производится повторная запись этого же фрагмента в формате стерео (двухканальная запись) с частотой дискретизации 64 кГц и 16-битным разрешением. Сжатие данных не производилось. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.
Ответ: _____.

**10** Вася и Петя передают друг другу сообщения, используя синий и красный фонарики. Эти они делают, включая фонарики последовательно на одинаково короткое время в некоторой комбинации. Количество вспышек в одном сообщении — от 1 до 5. Сколько различных сообщений могут передавать мальчики?
Ответ: _____.

**11** Алгоритм вычисления значения функции F($n$), где $n$ — натуральное число, задан следующими соотношениями:

F($n$) = F($n$ – 1) + F($n$/2), при $n$ > 1 и $n$ кратно 2

F($n$) = F($n$ – 1) + F($n$ – 2), при $n$ > 1 и $n$ не кратно 2

F(1) = 1

Чему равно значение функции F(8)?

*В ответе запишите только натуральное число.*

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

По заданным IP-адресу узла и маске определите адрес сети.

IP-адрес узла: 174.214.157.39

Маска: 255.255.240.0

При записи ответа выберите из приведённых в таблице чисел четыре элемента IP-адреса и запишите в нужном порядке соответствующие им буквы. Точки писать не нужно.

| A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| 0 | 39 | 144 | 157 | 174 | 214 | 240 | 255 |

*Пример.*

*Пусть искомый IP-адрес 192.168.128.0, и дана таблица*

| *A* | *B* | *C* | *D* | *E* | *F* | *G* | *H* |
|---|---|---|---|---|---|---|---|
| *128* | *168* | *255* | *8* | *127* | *0* | *17* | *192* |

*В этом случае правильный ответ будет записан в виде: HBAF*

Ответ: _____.

**13** Для регистрации на сайте онлайн-магазина пользователю предлагается придумать пароль, состоящий из 7 символов. Правилами безопасности сайта разрешается использовать только строчные буквы латинского алфавита и цифры в любом порядке. При этом используется посимвольное кодирование и в памяти сервера для кодирования каждого символа используется минимально возможное и одинаково целое количество бит. А для хранения всего пароля используется минимально возможное целое количество байт. Какое количество информации (в байтах) требуется для хранения паролей 40 пользователей?

Ответ: _____.

**14** Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости включает 4 команды-приказа и 4 команды проверки условия.

Команды-приказы:

| **вверх** | **вниз** | **влево** | **вправо** |
|---|---|---|---|

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно:

вверх ↑, вниз ↓, влево ←, вправо →.

Если РОБОТ начнёт движение в сторону находящейся рядом с ним стены, то он разрушится и программа прервётся.

Другие четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

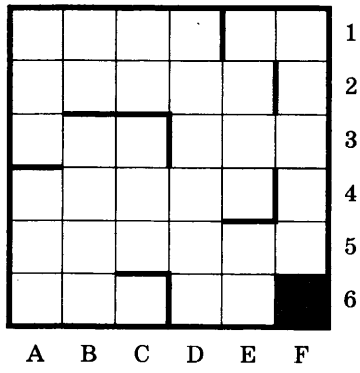
| **сверху свободно** | **снизу свободно** | **слева свободно** | **справа свободно** |
|---|---|---|---|

Цикл
ПОКА <**условие**>
**последовательность команд**
КОНЕЦ ПОКА
выполняется, пока условие истинно.

В конструкции
ЕСЛИ <**условие**>
ТО **команда1**
ИНАЧЕ **команда2**
КОНЕЦ ЕСЛИ
выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

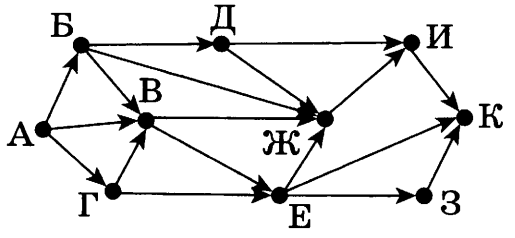
Сколько клеток лабиринта соответствует требованию, что, начав движение в ней и выполнив предложенную программу, РОБОТ уцелеет и остановится в закрашенной клетке (клетка F6)?

```
НАЧАЛО
ПОКА <справа свободно ИЛИ снизу свободно>
    ПОКА <справа свободно>
        вправо
    КОНЕЦ ПОКА
    влево
    ПОКА <снизу свободно>
        вниз
    КОНЕЦ ПОКА
    ЕСЛИ <справа свободно>
        вправо
    КОНЕЦ ЕСЛИ
КОНЕЦ ПОКА
КОНЕЦ
```




Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 25, запись которых в троичной системе счисления оканчивается на 21.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц |
|---|---|
| *Соус & (Кетчуп \| Молоко)* | 12300 |
| *Соус & Кетчуп & Молоко* | 0 |
| *Соус & Молоко* | 9700 |

Какое количество страниц будет найдено по запросу: *Соус & Кетчуп* ?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

**18** Обозначим через m&n поразрядную конъюнкцию неотрицательных целых чисел m и n. Так, например, $14\&5 = 1110_2 \,\&\, 0101_2 = 0100_2 = 4$.

Для какого наименьшего неотрицательного целого числа A формула:

$$(x\&A = 0 \wedge x\&36 = 0) \rightarrow x\&46 = 0$$

тождественно истинна (то есть принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 6; 5; 4; 7; 10; 6; 9; 11; 7; 8 соответственно, т.е. A[0] = 6; A[1] = 5 и т.д.

Определите значение переменной *k* после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
m = A[1]
FOR i = 1 TO 9
  IF A(i) > m THEN
    k = k + 1
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
m := A[1];
for i:=1 to 9 do
  if A[i] > m then
    k := k + 1;
```

**C++**

```
k = 0;
m = A[1];
for (i = 1 ; i <= 9 ; i++)
  if(A[i] > m)
    k = k + 1;
```

**Алгоритмический язык**

```
k := 0
m := A[1]
нц для i от 1 до 9
  если A[i] > m то
    k := k + 1
  все
кц
```

**Python**

```
k = 0
m = A[1]
for i in range(1,10):
    if A[i] > m:
        k = k + 1
```

Ответ: _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наименьшее из таких чисел *x*, при вводе которых алгоритм печатает сначала 0, а потом 24.

**Бейсик**

```
DIM X, M, L AS INTEGER
INPUT X
M = 1
L = 0
WHILE X > 0
  M = M * (X MOD 10)
  IF X MOD 10 > 5 THEN
    L = L + 1
  END IF
  X = X \ 10
WEND
PRINT L
PRINT M
```

**Паскаль**

```
var x, M, L: integer;
begin
  readln(x);
  M := 1;
  L := 0;
  while x > 0 do
  begin
    M := M * (x mod 10);
    if x mod 10 > 5 then
      L := L + 1;
    x := x div 10
  end;
  writeln(L); write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, M, L;
  cin >> x;
  M = 1; L = 0;
  while (x > 0) {
    M = M * (x % 10);
    if(x % 10 > 5)
      L = L + 1;
    x = x / 10;
  }
  cout << L << endl << M;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, M, L
  ввод x
  M := 1
  L := 0
  нц пока x > 0
    M := M * mod(x,10)
    если mod(x,10) > 5 то
      L := L + 1
    все
    x := div(x,10)
  кц
  вывод L, нс, M
кон
```

**Python**

```
x = int(input())
M = 1; L = 0
while x > 0:
    M = M * (x % 10)
    if x % 10 > 5:
        L = L + 1
    x = x // 10
print(L)
print(M)
```

**Ответ:** _____.

**21** **Определите, какое число будет напечатано в результате выполнения следующего алгоритма (для Вашего удобства алгоритм представлен на пяти языках):**

**Бейсик**

```
DIM A, B, T, M, R AS INTEGER
A = -15: B = 15
M = A:   R = F(A)
FOR T = A TO B
    IF F(T) < R THEN
       M = T
       R = F(T)
    END IF
NEXT T
PRINT M + R

FUNCTION F (x)
  F = (x*x - 16)*(x*x - 16) + 9
END FUNCTION
```

**Паскаль**

```
var a,b,t,M,R : integer;
function F(x:integer):integer;
begin
    F := (x*x-16)*(x*x-16) + 9
end;
BEGIN
 a := -15;  b := 15;
 M := a;    R := F(a);
 for t := a to b do
  if F(t) < R then
   begin
     M := t;
     R := F(t)
   end;
 write(M + R)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  return (x*x-16)*(x*x-16)+9;
}
int main() {
  long a, b, t, M, R;
  a = -15;  b = 15;
  M = a;    R = F(a);
  for (t = a; t <= b; t++)
    if ( F(t) < R ) {
      M = t;
      R = F(t);
    }
  cout << M + R << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел a, b, t, M, R
  a := -15; b := 15
  M := a;   R := F(a)
  нц для t от a до b
    если F(t) < R то
      M := t
      R := F(t)
    все
  кц
  вывод M + R
кон

алг цел F(цел x)
нач
  знач := (x*x-16)*(x*x-16) + 9
кон
```

**Python**

```
def F(x):
    return (x*x - 16)*(x*x - 16) + 9
a = -15; b = 15
M = a;   R = F(a)
for t in range(a,b+1):
    if (F(t) < R):
        M = t; R = F(t)
print(M + R)
```

Ответ: _____.

**22** У исполнителя Удвоитель две команды, которым присвоены номера:

**1. прибавь 1,**

**2. умножь на 2.**

Первая из них увеличивает число на экране на 1, вторая удваивает его.

Программа для Удвоителя — это последовательность команд.

Сколько есть программ, которые число 4 преобразуют в число 29?

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_9$, $x_{10}$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \to x_2) \wedge ((x_1 \equiv x_2) \to (x_3 \equiv x_4)) = 1$

$(x_3 \to x_4) \wedge ((x_3 \equiv x_4) \to (x_5 \equiv x_6)) = 1$

$(x_5 \to x_6) \wedge ((x_5 \equiv x_6) \to (x_7 \equiv x_8)) = 1$

$(x_7 \to x_8) \wedge ((x_7 \equiv x_8) \to (x_9 \equiv x_{10})) = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_9$, $x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры последовательность из $n$ целых чисел ($n = 4$), и выводит на экран произведение положительных чисел среди этой последовательности. Если в последовательности нет положительных чисел, программа выводит «NO». Известно, что вводимые числа не превышают по модулю 1000. Программист торопился и написал программу неправильно.

**Паскаль**

```
const n=4;
var i, a, prod, count: integer;
begin
 prod := 0;
 count := 0;
 for i := 1 to n do
 begin
   read(a);
   if a > 0 then
   begin
     prod := prod + a;
     count := count + 1
   end;
 end;
 if count > 0 then
   writeln(prod)
 else
   writeln('NO')
end.
```

**Бейсик**

```
CONST N=4
DIM I, A, PROD, COUNT AS INTEGER
PROD := 0
COUNT := 0
FOR I = 1 TO N
  INPUT A
  IF A > 0 THEN
    PROD = PROD + A
    COUNT = COUNT + 1
  END IF
NEXT I
IF COUNT > 0 THEN
  PRINT PROD
ELSE
  PRINT "NO"
END IF
END
```

**C++**

```
#include <iostream>
using namespace std;
#define n 4
int main() {
  int i, a, prod, count;
  prod = 0;
  count = 0;
  for (i = 0 ; i < n ; i++) {
    cin >> a;
    if(a > 0) {
      prod = prod + a;
      count = count + 1;
    }
  }
  if (count > 0)
    cout << prod << endl;
  else
    cout << "NO" << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел n = 4
  цел a, i, prod, count
  prod := 0
  count := 0
  нц для i от 1 до n
    ввод a
    если a > 0 то
      prod := prod + a
      count := count + 1
    все
  кц
  если count > 0 то
    вывод prod
  иначе
    вывод 'NO'
  все
кон
```

**Python**

```
N = 4
prod = 0
count = 0
for i in range(n):
    a = int(input())
    if a > 0:
        prod = prod + a
        count = count + 1
if count > 0:
    print(prod)
else:
    print("NO")
```

**Выполните следующие действия:**

**1. Напишите, что выведет программа при вводе чисел –1 2 –3 4.**

**2. Приведите пример такой входной последовательности, при которой программа работает верно.**

**3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.**

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать значения от –10 000 до 10 000. Опишите на одном из языков программирования алгоритм, который находит и выводит значение минимального трёхзначного положительного числа, которое не оканчивается на 4. Если такого элемента нет, вывести на экран «NO».

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of integer;
    i, k, m: integer;
begin
   for i:=1 to N do readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, K, M AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, k, m;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, k, m
  нц для i от 1 до N
    ввод a[i]
  кц
   ...
кон
```

**Python**

```
# допускается также
# использовать две
# целочисленные переменные k и m
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить** в кучу **один** или **два** камня или увеличить количество камней в куче в **три** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 17 или 45 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 30. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **30 или больше** камней.

В начальный момент в куче было S камней, $1 \leqslant S \leqslant 29$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. а) Укажите все такие значения числа S, при которых Петя может выиграть в один ход. Обоснуйте, что найдены все нужные значения S, и укажите выигрывающий ход для каждого указанного значения S.

б) Укажите такое значение S, при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом. Опишите выигрышную стратегию Вани.

2. Укажите два значения S, при которых у Пети есть выигрышная стратегия, причём (а) Петя не может выиграть за один ход и (б) Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Для каждого указанного значения S опишите выигрышную стратегию Пети.

3. Укажите такое значение S, при котором:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети, и

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом. Для указанного значения S опишите выигрышную стратегию Вани. Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

**27** На вход программе подаётся последовательность символов, заканчивающаяся символом #. Другие символы # во входной последовательности отсутствуют.

Программа должна вывести на экран символы латинского алфавита, в порядке увеличения частоты встречаемости во входной последовательности.

Если буква во входной последовательности не встречается, ее выводить не нужно.

Если несколько букв встречаются одинаковое количество раз, программа должна вывести их в алфавитном порядке.

Строчные и прописные буквы не различаются.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая должна решать поставленную задачу.

*Пример входных данных:*

```
Aced, ccedaa f#
```

*Пример выходных данных:*

```
FDEAC
```

# ВАРИАНТ 8

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Даны 4 целых числа, записанных различных системах счисления: $9F_{16}$, $10110110_2$, $A8_{16}$, $D1_{16}$. Сколько среди них чисел, значение которых лежит между $236_8$ и $B7_{16}$?

Ответ: _____.

**2** Вася заполнял таблицу истинности функции $w \wedge \neg x \wedge ((\neg y) \to \neg z)$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных $w$, $x$, $y$, $z$:

| | | | | $w \wedge \neg x \wedge ((\neg y) \to \neg z)$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

Определите, какому столбцу таблицы соответствует каждая из переменных $w$, $x$, $y$, $z$.

В ответе напишите буквы $w$, $x$, $y$, $z$ в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу и т.д.) Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

*Пример.* Если бы функция была задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имел бы вид

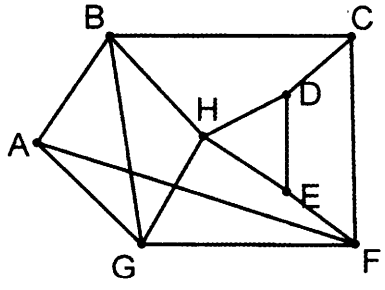
| | | $\neg x \vee y$ |
|---|---|---|
| 0 | 1 | 0 |

то первому столбцу соответствовала бы переменная $y$, а второму столбцу — переменная $x$. В ответе следовало бы написать $yx$.

Ответ: _____.

**3** На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о дорогах между населёнными пунктами (звёздочка означает, что дорога между соответствующими городами есть).

| | П1 | П2 | П3 | П4 | П5 | П6 | П7 | П8 |
|---|---|---|---|---|---|---|---|---|
| П1 | | | * | * | * | * | | |
| П2 | | | | * | | * | * | |
| П3 | * | | | | * | | | * |
| П4 | * | * | | | | | * | |
| П5 | * | | * | | | | * | * |
| П6 | * | * | | | | | | * |
| П7 | | * | | * | * | | | * |
| П8 | | | * | | * | * | * | |




Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, номера населённых пунктов **D** и **G** в таблице. В ответе напишите два числа без разделителей. Сначала для пункта **D**, затем для пункта **G**.

Ответ: _____.

**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите, сколько прямых потомков (т.е. детей и внуков) Калпен Л.В. упомянуто в таблице.

*Таблица 1*

| ID | Фамилия И.О. | Пол |
|---|---|---|
| 1518 | Гираб В.А. | М |
| 1885 | Жуйко А.И. | М |
| 2946 | Калпен Л.В. | М |
| 3472 | Арно В.А. | Ж |
| 4705 | Иваненко И.Л. | М |
| 5624 | Ирита Е.О. | Ж |
| 6109 | Кирта У.Т. | Ж |
| 7296 | Олинг К.Л. | Ж |
| 8434 | Кименко З.Т. | Ж |
| 9657 | Вирк Я.М. | Ж |
| | ... | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 1885 | 3472 |
| 6109 | 3472 |
| 9657 | 1885 |
| 4705 | 1885 |
| 2946 | 4705 |
| 5624 | 4705 |
| 8434 | 2946 |
| 1518 | 2946 |
| 2946 | 7296 |
| 5624 | 7296 |
| ... | ... |

Ответ: _____.

**5** Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, Д, Е, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовали кодовое слово 1; для буквы Б — кодовое слово 00. Какова наименьшая возможная сумма длин всех шести кодовых слов?

*Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: _____.

**6** Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются отдельно первая и вторая цифры, а также — вторая и третья цифры.
2. Полученные два числа записываются друг за другом в порядке невозрастания без разделителей.

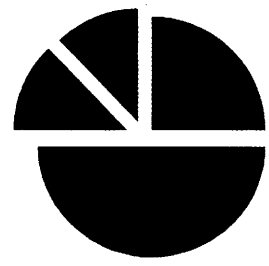
*Пример. Исходное число: 179. Суммы: 1 + 7 = 8; 7 + 9 = 16. Результат: 168.*

Укажите наибольшее число, при обработке которого автомат выдаёт результат 146.

Ответ: _____.

**7** Дан фрагмент электронной таблицы:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 3 | | 5 | |
| 2 | =(A1+C1)/4 | =C1-1 | =A2/2 | =B1/2 |



Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 3
s = 512
WHILE s > 1
  s = s \ 2
  k = k + 4
WEND
PRINT k
```

**Паскаль**

```
var k, s : integer;
begin
  k := 3; s := 512;
  while s > 1 do
  begin
    s := s div 2;
    k := k + 4
  end;
  write(k)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 3; s = 512;
   while (s > 1) {
     s = s / 2;
     k = k + 4;
   }
   cout << k << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 3
  s := 512
  нц пока s > 1
    s := div(s,2)
    k := k + 4
  кц
  вывод k
кон
```

**Python**

```
k = 3
s = 512
while s > 1:
    s = s // 2
    k = k + 4
print(k)
```

Ответ: _____.

**9** Производится двухканальная (стерео) звукозапись с частотой дискретизации 16 кГц и количеством уровней квантования 65536. Запись длится 4 минуты, её результаты записываются в файл, сжатие данных не производится. Укажите размер полученного файла (в Мбайт) с точностью 5 Мбайт. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

**10** Для того, чтобы пассажиры трамвая могли издалека видеть, трамвай какого маршрута приближается, в трамвайном парке применяют цветовое кодирование трамваев. Для этого на лобовом стекле используют светофильтры различных цветов — один слева и один справа. Известно, что светофильтры используются 5-ти различных цветов и что они не могут быть одинаковые на одном трамвае. Сколько различных цветовых кодов можно использовать?

Ответ: _____.

**11** Определите, сколько звёздочек будет напечатано в результате вызова F(3) приведённой подпрограммы:

**Бейсик**

```
SUB F(n)
  IF n > 0 THEN
    F(n - 2)
    F(n - 1)
    F(n - 1)
  END IF
  PRINT "*";
END SUB
```

**Паскаль**

```
procedure F(n: integer);
begin
  if n > 0 then
  begin
    F(n - 2);
    F(n - 1);
    F(n - 1)
  end;
  write('*')
end;
```

**C++**

```
void F(int n) {
  if (n > 0) {
    F(n - 2);
    F(n - 1);
    F(n - 1);
  }
  std::cout << "*";
}
```

**Алгоритмический язык**

```
алг F(цел n)
нач
  если n > 0 то
    F(n - 2)
    F(n - 1)
    F(n - 1)
  все
  вывод '*'
кон
```

**Python**

```
def F(n):
    if n > 0:
        F(n - 2)
        F(n - 1)
        F(n - 1)
    print("*")
```

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

*Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.*

Для узла с IP-адресом 243.212.83.98 адрес сети равен 243.212.80.0. Чему равно наименьшее возможное количество единиц в разрядах маски?

Ответ: _____.

**13** При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 13 символов и содержащий только символы из 15-символьного набора: A, B, C, D, E, F, G, H, K, L, M, N, O, P, Q. В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего отведено 8 байт на одного пользователя.

Определите объём памяти (в байтах), необходимый для хранения сведений о 20 пользователях. В ответе запишите только целое число — количество байт.

Ответ: _____.

**14** Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

**А) заменить (v, w).**

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w. Например, выполнение команды **заменить (222, 34)** преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v, то выполнение команды **заменить (v, w)** не меняет эту строку.

**Б) нашлось (v).**

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл
**ПОКА условие**
**последовательность команд**
**КОНЕЦ ПОКА**
выполняется, пока условие истинно.

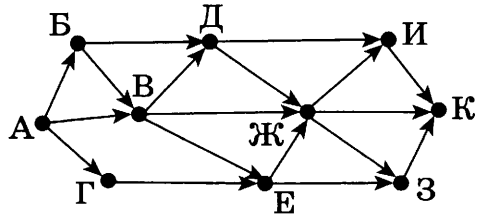
В конструкции
**ЕСЛИ условие**
**ТО команда1**
**ИНАЧЕ команда2**
**КОНЕЦ ЕСЛИ**
выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 65 идущих подряд цифр 4? В ответе запишите полученную строку.

```
НАЧАЛО
    ПОКА нашлось (444) ИЛИ нашлось (555)
        ЕСЛИ нашлось (444)
          ТО заменить (444, 5)
          ИНАЧЕ заменить (555, 4)
        КОНЕЦ ЕСЛИ
    КОНЕЦ ПОКА
КОНЕЦ
```

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Значение арифметического выражения: $25^{40} - 5^{30} + 24$ — записали в системе счисления с основанием 5. Сколько цифр «4» содержится в этой записи?

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| *Корвет \| Линкор \| Фрегат* | 30 |
| *Фрегат* | 17 |
| *Линкор* | 12 |
| *Корвет* | 8 |
| *Линкор & Фрегат* | 4 |
| *Корвет & Линкор* | 3 |
| *Корвет & Линкор & Фрегат* | 2 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу *Корвет & Фрегат* ?

Ответ: _____.

**18** Для какого наибольшего целого числа $A$ формула $(xy > A) \vee (x < y) \vee (y \leqslant 9)$ тождественно истинна, то есть принимает значение 1 при любых целых неотрицательных $x$ и $y$?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 6; 5; 4; 7; 10; 6; 9; 11; 7; 8 соответственно, т.е. A[0] = 6; A[1] = 5 и т.д.

Определите значение переменной *k* после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
m = 0
FOR i = 0 TO 9
  IF A(i) > m THEN
    k = k + 1
    m = A(i)
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
m := 0;
for i := 0 to 9 do
  if A[i] > m then
  begin
    k := k + 1;
    m := A[i]
  end;
```

**C++**

```
k = 0;
m = 0;
for (i=0 ; i<=9 ; i++)
  if(A[i] > m) {
    k = k + 1;
    m = A[i];
  }
```

**Алгоритмический язык**

```
k := 0
m := 0
нц для i от 0 до 9
  если A[i] > m то
    k := k + 1
    m := A[i]
  все
кц
```

**Python**

```
k = 0
m = 0
for i in range(10):
    if A[i] > m:
        k = k + 1
        m = A[i]
```

**Ответ:** _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число *x*, этот алгоритм печатает два числа *L* и *M*. Укажите наименьшее из таких чисел *x*, при вводе которых алгоритм печатает сначала 3, а потом 5.

**Бейсик**

```
DIM X, L, M AS INTEGER
INPUT X
L = 0
M = 9
WHILE X > 0
  L = L + 1
  IF M > (X MOD 10) THEN
    M = X MOD 10
  END IF
  X = X \ 10
WEND
PRINT L
PRINT M
```

**Паскаль**

```
var x, L, M: integer;
begin
  readln(x);
  L := 0; M := 9;
  while x > 0 do
   begin
    L := L + 1;
    if M > (x mod 10) then
       M := x mod 10;
    x := x div 10
   end;
  writeln(L); write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, L, M;
  cin >> x;
  L = 0; M = 9;
  while (x > 0) {
    L = L + 1;
    if (M > x % 10)
      M = x % 10;
    x = x / 10;
  }
  cout << L << endl << M;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, L, M
  ввод x
  L := 0
  M := 9
  нц пока x > 0
    L := L + 1
    если M > mod(x,10) то
      M := mod(x,10)
    все
    x := div(x,10)
  кц
  вывод L, нс, M
кон
```

**Python**

```
x = int(input())
L = 0; M = 9
while x > 0:
    L = L + 1
    if M > x % 10:
        M = x % 10
    x = x // 10
print(L)
print(M)
```

**Ответ:** _____.

**21** Определите, при каком наименьшем значении b в результате выполнения следующего алгоритма будет напечатано число 15 (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B AS INTEGER
INPUT B
A = 0
WHILE F(A) < B
  A = A + 1
WEND
PRINT A

FUNCTION F (x)
  IF x = 0
    F = 0
  ELSE
    F = F(x - 1) + 8
  END IF
END FUNCTION
```

**Паскаль**

```
var a,b : integer;
function F(x:integer):integer;
begin
  if x = 0 then
    F := 0
  else
    F := F(x - 1) + 8
end;
BEGIN
 readln(b);
 a := 0;
 while F(a) < b do
   a := a + 1;
 write(a)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  if(x == 0)
    return 0;
  else
    return F(x - 1) + 8;
}
int main() {
  long a, b;
  cin >> b;
  a = 0;
  while (F(a) < b)
    a = a + 1;
  cout << a << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел a, b
 ввод b
 a := 0
 нц пока F(a) < b
   a := a + 1
 кц
 вывод a
кон

алг цел F(цел x)
нач
 если x = 0 то
   знач := 0
 иначе
   знач := F(x - 1) + 8
 все
кон
```

**Python**

```
def F(x):
    if x == 0:
        return 0
    else:
        return F(x - 1) + 8
b = int(input())
a = 0
while F(a) < b:
    a = a + 1
print(a)
```

**Ответ:** _____.

**22** У исполнителя Прибавлятель две команды, которым присвоены номера:
**1. прибавь 1,**
**2. прибавь 9.**
Первая из них увеличивает число на экране на 1, вторая увеличивает его на 9.
Программа для Прибавлятеля — это последовательность команд.
Сколько есть программ, которые число **2** преобразуют в число **25**?
Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_7$, $x_8$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \to x_2) \vee (x_1 \to x_3) = 1$
$(x_2 \to x_3) \vee (x_2 \to x_4) = 1$
$(x_3 \to x_4) \vee (x_3 \to x_5) = 1$
$(x_4 \to x_5) \vee (x_4 \to x_6) = 1$
$(x_5 \to x_6) \vee (x_5 \to x_7) = 1$
$(x_6 \to x_7) \vee (x_6 \to x_8) = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_7$, $x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.
Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры последовательность из $n$ целых чисел ($n = 4$), и выводит на экран минимальное положительное число этой последовательности. Если в последовательности нет положительных чисел, программа должна вывести на экран «NO». Известно, что вводимые числа не превышают по модулю 1000. Программист торопился и написал программу неправильно.

**Паскаль**

```
const n=4;
var i, a, min1: integer;
begin
 min1 := 2000;
 for i := 1 to n do
 begin
   read(a);
   if a > 0 then
     if a < min1 then
       min1 := a
 end;
 if min1 > 0 then
   writeln(min1)
 else
   writeln(0)
end.
```

**Бейсик**

```
CONST N=4
DIM I, A, MIN1 AS INTEGER
MIN1 := 2000
FOR I = 1 TO N
  INPUT A
  IF A > 0 THEN
    IF A < MIN1 THEN
      MIN1 = A
    END IF
  END IF
NEXT I
IF MIN1 > 0 THEN
  PRINT MIN1
ELSE
  PRINT 0
END IF
END
```

**C++**

```
#include <iostream>
using namespace std;
const int n = 4;
int main() {
  int i, a, min1;
  min1 = 2000;
  for (i=0 ; i < n ; i++) {
    cin >> a;
    if(a > 0)
      if(a < min1)
        min1 = a;
  }
  if(min1 > 0)
    cout << min1 << endl;
  else
    cout << 0 << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел n = 4
  цел a, i, min1
  min1 := 2000
  нц для i от 1 до n
    ввод a
    если a > 0 то
      если a < min1 то
        min1 := a
      все
    все
  кц
  если min1 > 0 то
    вывод min1
  иначе
    вывод 0
  все
кон
```

**Python**

```
N = 4
min1 = 2000
for i in range(n):
    a = int(input())
    if a > 0:
        if a < min1:
            min1 = a
if min1 > 0:
    print(min1)
else:
    print(0)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел –1 –2 –3 –4.

2. Приведите пример такой входной последовательности, при которой программа работает верно.

3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать значения от –10 000 до 10 000. Опишите на одном из языков программирования алгоритм, который находит и выводит количество пар соседних элементов, произведение которых чётно и положительно.

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of integer;
    i, k, p: integer;
begin
   for i:=1 to N do
     readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, K, P AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, k, p;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, k, p
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

**Python**

```
# допускается также
# использовать две
# целочисленные переменные k и p
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26**

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 47. Победителем считается игрок, сделавший последний ход, то есть первым получивший такую позицию, что в кучах всего будет 47 или больше камней.

В начальный момент в первой куче было 5 камней, во второй куче — S камней;

$$1 \leqslant S \leqslant 41.$$

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, то есть не являющиеся выигрышными независимо от игры противника.

Выполните следующие задания.

**Задание 1**

а) Укажите все такие значения числа S, при которых Петя может выиграть за один ход.

б) Укажите наименьшее значение S, при котором Ваня может выиграть своим первым ходом при хотя бы каком-нибудь первом ходе Пети?

**Задание 2**

Укажите такое значение S, при котором у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

– Петя не может выиграть за один ход;

– Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Для указанного значения S опишите выигрышную стратегию Пети.

**Задание 3**

Укажите значение S, при котором одновременно выполняются два условия:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Для указанного значения S опишите выигрышную стратегию Вани.

Постройте дерево всех партий, возможных при этой выигрышной стратегии Вани (в виде рисунка или таблицы). На рёбрах дерева указывайте ходы, в узлах указывайте позиции.

**27**

После единых выпускных экзаменов по информатике в район пришла информация о том, какой ученик какой школы сколько баллов набрал.

Районный методист решила выяснить номер школы, ученики которой набрали наибольший средний балл, с точностью до целых.

Программа должна вывести на экран номер такой школы и её средний балл.

Если наибольший средний балл набрало больше одной школы — вывести количество таких школ.

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая должна вывести на экран требуемую информацию. Известно, что информатику сдавало больше 5-ти учеников района. Также известно, что в районе школы с некоторыми номерами не существуют.

На вход программе сначала подаётся число учеников, сдававших экзамен. В каждой из следующих N строк находится информация об учениках в формате:

<Фамилия> <Имя> <Номер школы> <Количество баллов>

где <Фамилия> — строка, состоящая не более, чем из 30 символов без пробелов, <Имя> — строка, состоящая не более, чем из 20 символов без пробелов, <Номер школы> — целое число в диапазоне от 1 до 99, <Количество баллов> — целое число диапазоне от 1 до 100. Эти данные записаны через пробел, причём ровно один между каждой парой (то есть, всего по три пробела в каждой строке).

*Пример входной строки:*

```
Иванов Иван 50 87
```

*Пример выходных данных:*

```
50 74
```

*Другой вариант выходных данных:*

```
7
```

# ВАРИАНТ 9

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Сколько чисел находится между $C9_{16}$ и $D4_{16}$?

Ответ: _____.

**2** Дан фрагмент таблицы истинности выражения F. Значения в пустых клетках таблицы неизвестны и могут быть любыми из возможных.

| $x1$ | $x2$ | $x3$ | F |
|---|---|---|---|
| | 1 | 0 | 0 |
| 0 | | 0 | 0 |
| 1 | 0 | | 1 |

Перечислите в порядке возрастания без запятых и пробелов номера логических выражений, которые могут соответствовать F:

1) $\neg x1 \vee x2 \vee \neg x3$
2) $x1 \wedge \neg x2 \vee x3$
3) $x1 \wedge \neg x2 \wedge \neg x3$
4) $x1 \vee \neg x2 \vee x3$
5) $x1 \wedge x2 \wedge x3$

Ответ: _____.

**3** Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | | 9 | 6 | 3 | | |
| B | 9 | | 3 | | 2 | |
| C | 6 | 3 | | 2 | 7 | 8 |
| D | 3 | | 2 | | | 10 |
| E | | 2 | 7 | | | 2 |
| F | | | 8 | 10 | 2 | |

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

Ответ: _____.

**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите, у какого количества человек, упомянутых в таблице, есть сестра, возраст которой отличается не более чем на 5 лет.

*Таблица 1*

| ID | Фамилия И.О. | Пол | Год |
|---|---|---|---|
| 1518 | Гираб В.А. | М | 1987 |
| 1885 | Жуйко А.И. | М | 1990 |
| 2458 | Омни Е.А. | М | 2010 |
| 2946 | Калпен Л.В. | М | 1937 |
| 3472 | Арно В.А. | Ж | 2014 |
| 4705 | Иваненко И.Л. | М | 1972 |
| 5624 | Ирита Е.О. | Ж | 1942 |
| 6109 | Кирта У.Т. | Ж | 1995 |
| 7296 | Олинг К.Л. | Ж | 1970 |
| 8434 | Кименко З.Т. | Ж | 1966 |
| 9657 | Вирк Я.М. | Ж | 1965 |
| | ... | | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 1885 | 3472 |
| 6109 | 3472 |
| 9657 | 1885 |
| 4705 | 1885 |
| 2946 | 4705 |
| 5624 | 4705 |
| 2946 | 7296 |
| 5624 | 7296 |
| 2946 | 8434 |
| 5624 | 8434 |
| 1885 | 2458 |
| 6109 | 2458 |
| 9657 | 1518 |
| 4705 | 1518 |

Ответ: _____.

**5** Для передачи информации по каналу связи с помехами используется 5-битовый код. Сообщение содержит только буквы A, B и C, которые кодируются следующими кодовыми словами:

A-01101, B-11000, C-10110.

Любые два из этих трёх кодовых слов отличаются не менее чем в трёх позициях. Поэтому, если принятое кодовое слово отличается от допустимого не более чем в одной позиции, можно однозначно определить, какая буква передавалась (говорят, что «код исправляет одну ошибку»). Так, получив кодовое слово 10 000, можно догадаться, что передавалась буква B (отличие от кодового слова для B только в одной позиции, для остальных кодовых слов отличий больше). Если принятое кодовое слово отличается от всех допустимых кодовых слов более, чем в одной позиции, считается, что произошла ошибка (она обозначается «x»)

Получено сообщение 01001 00011 10101 11001. Декодируйте это сообщение. В ответе укажите четыре символа без пробелов и запятых.

Ответ: _____.

**6** Автомат получает на вход трёхзначное число. По этому числу строится новое число по следующим правилам.

1. Складываются отдельно первая и вторая цифры, а также — вторая и третья цифры.
2. Полученные два числа записываются друг за другом в порядке неубывания без разделителей.

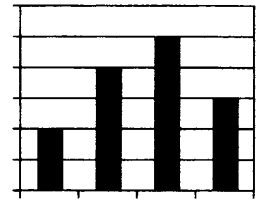
*Пример. Исходное число: 179. Суммы: 1 + 7 = 8; 7 + 9 = 16. Результат: 816.*

Укажите наибольшее число, при обработке которого автомат выдаёт результат 710.

Ответ: _____.

**7** Дан фрагмент электронной таблицы:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 3 | | 5 | |
| 2 | =C1-A1 | =A2*2 | =A1+2 | =B1-B2 |



Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 90
s = 0
WHILE s + k > 20
  s = s + 5
  k = k - 15
WEND
PRINT s
```

**Паскаль**

```
var k, s : integer;
begin
  k := 90;
  s := 0;
  while s + k > 20 do
  begin
    s := s + 5;
    k := k - 15
  end;
  write(s)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 90;
   s = 0;
   while (s + k > 20) {
     s = s + 5;
     k = k - 15;
   }
   cout << s << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 90
  s := 0
  нц пока s + k > 20
    s := s + 5
    k := k - 15
  кц
  вывод s
кон
```

**Python**

```
k = 90
s = 0
while s + k > 20:
    s = s + 5
    k = k - 15
print(s)
```

Ответ: _____.

**9** Документ объёмом 16 Мбайт можно передать с одного компьютера на другой двумя способами:

А. Сжать архиватором, передать по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и на сколько, если:

- средняя скорость передачи данных по каналу связи составляет $2^{21}$ бит в секунду;
- объём сжатого архиватором документа равен 25% исходного;
- время, требуемое на сжатие документа, — 8 секунд, на распаковку — 3 секунды?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ: _____.

**10** В некотором племени для общения используется всего 64 слова. Вождь племени на общем собрании произносит 32 слова. Какое количество информации (число бит) сообщает вождь?

Ответ: _____.

**11** Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(5).

**Бейсик**

```
SUB F(n)
  IF n > 0 THEN
    PRINT n;
    F(n \ 3)
    F(n - 1)
  END IF
END SUB
```

**Паскаль**

```
procedure F(n: integer);
begin
  if n > 0 then
  begin
    write(n);
    F(n div 3);
    F(n - 1)
  end
end;
```

**C++**

```
void F(int n) {
  if (n > 0) {
    std::cout << n;
    F(n / 3);
    F(n - 1);
  }
}
```

**Алгоритмический язык**

```
алг F(цел n)
нач
  если n > 0 то
    вывод n
    F(div(n,3))
    F(n - 1)
  все
кон
```

**Python**

```
def F(n):
    if n > 0:
        print(n)
        F(n // 3)
        F(n - 1)
```

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули.

Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске.

*Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.*

Для узла с IP-адресом 117.203.214.56 адрес сети равен 117.203.192.0. Чему равно наибольшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: _____.

**13** На стадионе имеется 10 секторов, в каждом из которых установлено по 80 кресел в 40 рядов. При кодировании номера места автоматизированная система продажи билетов использует минимально возможное количество бит, одинаковое для каждого номера места, отдельно номер сектора, номер ряда и номер места в ряду. При этом для записи кода используется минимально возможное и одинаково целое количество байтов. Каков объём информации (в байтах), записанный устройством после продажи 200 билетов?

Ответ: _____.

**14** Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

**А) заменить (v, w).**

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w. Например, выполнение команды **заменить (222, 34)** преобразует строку 77222277 в строку 7734277.

Если в строке нет вхождений цепочки v, то выполнение команды **заменить (v, w)** не меняет эту строку.

**Б) нашлось (v).**

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

**ПОКА условие**
**последовательность команд**
**КОНЕЦ ПОКА**

выполняется, пока условие истинно.

В конструкции

**ЕСЛИ условие**
**ТО команда1**
**ИНАЧЕ команда2**
**КОНЕЦ ЕСЛИ**

выполняется **команда1** (если условие истинно) или **команда2** (если условие ложно).

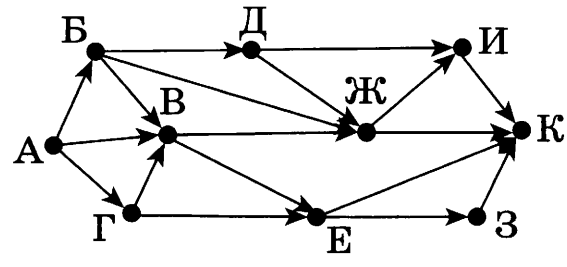
Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 69 идущих подряд цифр 7? В ответе запишите полученную строку.

```
НАЧАЛО
  ПОКА нашлось (777) ИЛИ нашлось (888)
      ЕСЛИ нашлось (888)
        ТО заменить (888, 7)
        ИНАЧЕ заменить (777, 8)
      КОНЕЦ ЕСЛИ
  КОНЕЦ ПОКА
КОНЕЦ
```

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей из города А в город К?




Ответ: _____.

**16** Какое восьмеричное число находится ровно посередине между числами $C8_{16}$ и $11010100_2$? В ответе напишите только само число в 8-ричной системе счисления. Основание системы счисления писать не нужно.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| *Бирка & Табличка & Ценник* | 5 |
| *Бирка* | 36 |
| *Табличка* | 43 |
| *Ценник* | 48 |
| *Бирка & Табличка* | 18 |
| *Бирка & Ценник* | 20 |
| *Табличка & Ценник* | 28 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу *Бирка | Табличка | Ценник*?

Ответ: _____.

**18** Укажите наименьшее целое значение $A$, при котором выражение

$$(x > 30) \vee (y > 40) \vee (2x + 3y < A)$$

истинно для любых целых значений $x$ и $y$.

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 6; 3; 9; 11; 7; 8 соответственно, т.е. A[0] = 4; A[1] = 5 и т.д.

Определите значение переменной k после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
m = 0
FOR i = 1 TO 9
  IF A(i) > m THEN
    m = A(i)
    k = k + 1
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
m := A[0];
for i := 1 to 9 do
  if A[i] > m then
  begin
    m := A[i];
    k := k + 1;
  end;
```

**C++**

```
k = 0;
m = A[0];
for (i=1 ; i<=9 ; i++)
  if (A[i] > m) {
    m = A[i];
    k = k + 1;
  }
```

**Алгоритмический язык**

```
k := 0
m := A[0]
нц для i от 1 до 9
  если A[i] > m то
      m := A[i]
      k := k + 1
  все
кц
```

**Python**

```
k = 0
m = A[0]
for i in range(1,10):
    if A[i] > m:
        m = A[i]
        k = k + 1
```

Ответ: _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число $x$, этот алгоритм печатает два числа: $a$ и $b$. Укажите **наименьшее** из таких чисел $x$, при вводе которого алгоритм печатает сначала 6, а потом 9.

**Бейсик**

```
DIM X, A, B AS INTEGER
INPUT X
A = 1 : B = 0
WHILE X > 0
  IF X MOD 2 = 0 THEN
    A = A * (X MOD 8)
  ELSE
    B = B + X MOD 8
  END IF
  X = X \ 8
WEND
PRINT A
PRINT B
```

**Паскаль**

```
var x, a, b: integer;
begin
 readln(x);
 a := 1; b := 0;
 while x > 0 do begin
   if x mod 2 = 0 then
     a := a * (x mod 8)
   else
     b := b + x mod 8;
   x := x div 8
 end;
 writeln(a); write(b)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, a, b;
  cin >> x;
  a = 1; b = 0;
  while (x > 0) {
    if(x % 2 == 0)
      a = a * (x % 8);
    else
      b = b + x % 8;
    x = x / 8;
  }
  cout << a << endl << b;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, a, b
  ввод x
  a := 1
  b := 0
  нц пока x > 0
    если mod(x,2) = 0 то
      a := a * mod(x,8)
    иначе
      b := b + mod(x,8)
    все
    x := div(x,8)
  кц
  вывод a, нс, b
кон
```

**Python**

```
x = int(input())
a = 1; b = 0
while x > 0:
    if x % 2 == 0:
        a = a * (x % 8)
    else:
        b = b + x % 8
    x = x // 8
print(a)
print(b)
```

**Ответ:** _____.

**21** Определите, при каком наибольшем значении b в результате выполнения следующего алгоритма будет напечатано число 10 (для Вашего удобства алгоритм представлен на пяти языках):

**Бейсик**

```
DIM A, B AS INTEGER
INPUT B
A = 0
WHILE A < F(B)
  A = A + 1
WEND
PRINT A

FUNCTION F (x)
  IF x < 7
    F = 0
  ELSE
    F = F(x - 7) + 1
  END IF
END FUNCTION
```

**Паскаль**

```
var a,b : integer;
function F(x:integer):integer;
begin
  if x < 7 then
    F := 0
  else
    F := F(x - 7) + 1
end;
BEGIN
 readln(b);
 a := 0;
 while a < F(b) do
   a := a + 1;
 write(a)
END.
```

**C++**

```
#include <iostream>
using namespace std;
long F(long x) {
  if(x < 7)
    return 0;
  else
    return F(x - 7) + 1;
}
int main() {
  long a, b;
  cin >> b;
  a = 0;
  while (a < F(b))
    a = a + 1;
  cout << a << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел a, b
 ввод b
 a := 0
 нц пока a < F(b)
   a := a + 1
 кц
 вывод a
кон

алг цел F(цел x)
нач
 если x < 7 то
   знач := 0
 иначе
   знач := F(x - 7) + 1
 все
кон
```

**Python**

```
def F(x):
    if x < 7:
        return 0
    else:
        return F(x - 7) + 1
b = int(input())
a = 0
while a < F(b):
    a = a + 1
print(a)
```

Ответ: _____.

**22** Исполнитель Июнь12 преобразует число на экране.

У исполнителя есть две команды, которым присвоены номера:

**1. Прибавить 3**

**2. Умножить на 2**

Первая команда увеличивает число на экране на 3, вторая умножает его на 2.

Программа для исполнителя Июнь12 — это последовательность команд.

Сколько существует программ, для которых при исходном числе **3** результатом является число **45** и при этом траектория вычислений содержит число **18**?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 10, 20, 23.

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, ..., $x_7$, $x_8$, которые удовлетворяют всем перечисленным ниже условиям?

$(x_1 \equiv x_2) \vee (\neg x_1 \wedge x_3) \vee (x_1 \wedge \neg x_3) = 1$
$(x_2 \equiv x_3) \vee (\neg x_2 \wedge x_4) \vee (x_2 \wedge \neg x_4) = 1$
$(x_3 \equiv x_4) \vee (\neg x_3 \wedge x_5) \vee (x_3 \wedge \neg x_5) = 1$
$(x_4 \equiv x_5) \vee (\neg x_4 \wedge x_6) \vee (x_4 \wedge \neg x_6) = 1$
$(x_5 \equiv x_6) \vee (\neg x_5 \wedge x_7) \vee (x_5 \wedge \neg x_7) = 1$
$(x_6 \equiv x_7) \vee (\neg x_6 \wedge x_8) \vee (x_6 \wedge \neg x_8) = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, ..., $x_7$, $x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24** Требовалось написать программу, которая вводит с клавиатуры последовательность целых ненулевых чисел, оканчивающуюся нулём, и выводит на экран среднее арифметическое нечётных чисел среди этой последовательности. Если в последовательности нет нечётных чисел, программа должна вывести «NO». Известно, что последовательность не пуста. Программист торопился и написал программу неправильно.

**Паскаль**

```
var a, s, c: integer;
begin
 s := 0;
 c := 0;
 read(a);
 while a <> 0 do
 begin
   if a mod 2 = 0 then
   begin
     s := s + a;
     c := c + 1
   end;
   read(a)
 end;
 if s = 0 then
   writeln('NO')
 else
   writeln(s/c)
end.
```

**Бейсик**

```
s = 0
c = 0
INPUT a
WHILE a <> 0
  IF a MOD 2 = 0 THEN
    s = s + a
    c = c + 1
  ENDIF
  INPUT a
WEND
IF s = 0 THEN
  PRINT "NO"
ELSE
  PRINT s / c
ENDIF
END
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int a, s;
  s = 0;
  c = 0;
  cin >> a;
  while (a != 0) {
    if(a % 2 == 0) {
      s = s + a;
      c = c + 1;
    }
    cin >> a;
  }
  if (s == 0)
    cout << "NO" << endl;
  else
    cout << (float)s / c << endl;
}
```

**Алгоритмический язык**

```
алг
нач
  цел a, s
  s := 0
  c := 0
  ввод a
  нц пока a <> 0
    если mod(a,2) = 0 то
      s := s + a
      c := c + 1
    все
    ввод a
  кц
  если s = 0 то
    вывод 'NO'
  иначе
    вывод s / c
  все
кон
```

**Python**

```
s = 0
c = 0
a = int(input())
while a != 0:
    if a % 2 == 0:
        s = s + a
        c = c + 1
    a = int(input())
if s == 0:
    print("NO")
else:
    print(s/c)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел 1 2 3 4 0.

2. Приведите пример такой входной последовательности, при которой программа работает верно.

3. Укажите все ошибки в программе и исправьте их. Для этого для каждой ошибки выпишите строку, которая написана неправильно и приведите правильный вариант строки.

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от –10 000 до 10 000 включительно.

Опишите на одном из языков программирования алгоритм, который находит наименьшее значение массива, которое кратно 4-м, и при этом не больше 100 по модулю, а затем заменяет каждый элемент массива, не удовлетворяющий данному условию, на найденное значение. Гарантируется, что хотя бы один искомый элемент в массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строчки.

Например, для массива из 9-ти элементов: `427 8 300 -52 -1234 65 -66 72 145`

программа должна вывести числа `-52 8 -52 -52 -52 -52 -52 72 -52`

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования.

Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

**Паскаль**

```
const N = 40;
var a: array [1..N] of integer;
    i, k, m : integer;
begin
   for i:=1 to N do
     read(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, K, M AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
#include <cstdlib>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, k, m;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, k, m
  нц для i от 1 до N
    ввод a[i]
  кц
   ...
кон
```

**Python**

```
# допускается также использовать
# две целочисленные переменные k и m
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26** Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит две кучи камней. Игроки ходят по очереди, первый ход делает Петя. Назовём позицией игры указание количества камней в обеих кучах. Например, позиция (15,20) означает, что в первой куче 15 камней, а во второй — 20. За один ход игрок может добавить в одну из куч **один** камень или увеличить количество камней в одной из куч в **два** раза. Например, имея количество камней в кучах (15,20), за один ход можно получить положение (16,20), (30,20), (15,21) и (15,30). У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в обеих кучах (сумма) становится не менее 65. Победителем считается игрок, сделавший последний ход, то есть первым получивший положение, в котором сумма камней в кучах будет **65 или больше** камней.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока — значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (4,31), (5,30), (7,29) выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Выполните следующие задания. Во всех случаях обосновывайте свой ответ.

1. Для исходных позиций (4,30) и (6,29) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

2. Для исходных позиций (4,29), (6,28) и (5,29) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

3. Для исходной позиции (5,28) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Приведите дерево игры. Обоснуйте ответ. На рёбрах дерева указывайте, кто делает ход, в узлах — позицию игры.

**27** После единых выпускных экзаменов по информатике в район пришла информация о том, какой ученик какой школы сколько баллов набрал. По положению об экзамене каждый район сам определяет, за какой балл нужно поставить какую оценку.

Районный методист решила, что оценку «отлично» должны получить 20% участников (целое число, с отбрасыванием дробной части).

Для этого она должна определить, какой балл должен был набрать ученик, чтобы получить «отлично».

Если невозможно определить такой балл, чтобы «отлично» получили ровно 20% участников, «отлично» должно получить меньше участников, чем 20%.

Если таких участников не окажется (наибольший балл набрали больше 20% участников) — эти и только эти ученики должны получить «отлично».

Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая должна вывести на экран наименьший балл, который набрали участники, получившие «отлично». Известно, что информатику сдавало больше 5-ти учеников. Также известно, что есть такое количество баллов, которое не получил ни один участник.

На вход программе сначала подаётся число учеников, сдававших экзамен. В каждой из следующих N строк находится информация об учениках в формате:

<Фамилия> <Имя> <Номер школы> <Количество баллов>

где <Фамилия> — строка, состоящая не более, чем из 30 символов без пробелов, <Имя> — строка, состоящая не более, чем из 20 символов без пробелов, <Номер школы> — целое число в диапазоне от 1 до 99, <Количество баллов> — целое число диапазоне от 1 до 100. Эти данные записаны через пробел, причём ровно один между каждой парой (то есть, всего по три пробела в каждой строке).

*Пример входной строки:*

```
Иванов Иван 50 87
```

*Пример выходных данных:*

78

# ВАРИАНТ 10

## Часть 1

*Ответами к заданиям 1–23 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.*

**1** Во сколько раз $101110000_2$ больше чем $10111_2$? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

**2** Дан фрагмент таблицы истинности выражения F. Значения в пустых клетках таблицы неизвестны и могут быть любыми из возможных.

| $x1$ | $x2$ | $x3$ | F |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 0 | | 1 |

Перечислите в порядке возрастания без запятых и пробелов номера логических выражений, которые могут соответствовать F:

1) $x1 \wedge \neg x2 \wedge x3$
2) $\neg x1 \vee x2 \vee x3$
3) $\neg x1 \vee \neg x2 \vee x3$
4) $x1 \wedge \neg x2 \wedge \neg x3$
5) $x1 \vee x2 \vee x3$

Ответ: _____.

**3** Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | | 3 | 8 | | | |
| B | 3 | | 6 | | 8 | |
| C | 8 | 6 | | 3 | | 4 |
| D | | | 3 | | 1 | 3 |
| E | | 8 | | 1 | | 2 |
| F | | | 4 | 3 | 2 | |

Определите длину кратчайшего пути между пунктами A и F (при условии, что передвигаться можно только по построенным дорогам).

Ответ: _____.

**4** Дан фрагмент базы данных, содержащий информацию о родственных отношениях. Определите количество человек, у которых есть внук или внучка, родившаяся ранее, чем через 50 лет от даты их рождения.

*Таблица 1*

| ID | Фамилия И.О. | Пол | Год |
|---|---|---|---|
| 1453 | Дени К.Д. | Ж | 1960 |
| 2185 | Ирта О.Б. | Ж | 1985 |
| 2605 | Гано И.Е. | Ж | 1995 |
| 3474 | Ивенко Д.Я. | М | 1944 |
| 4218 | Бабенко А.Е. | М | 2012 |
| 5363 | Марченко Е.Е. | М | 1990 |
| 5771 | Арне А.А. | М | 1937 |
| 7148 | Камо Е.А. | М | 1965 |
| 7814 | Исава Г.А. | Ж | 1941 |
| 8208 | Сорт А.К. | Ж | 2016 |
| | ... | | |

*Таблица 2*

| ID_Родителя | ID_Ребёнка |
|---|---|
| 5363 | 4218 |
| 2185 | 4218 |
| 7148 | 5363 |
| 1453 | 5363 |
| 3474 | 1453 |
| 7814 | 1453 |
| 7148 | 2605 |
| 1453 | 2605 |
| 2605 | 8208 |
| 5771 | 7814 |
| ... | ... |

Ответ: _____.

**5** Для кодирования некоторой последовательности, состоящей только из пяти букв А, Б, В, Г, Д, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовали кодовое слово 00; для буквы Б — кодовое слово 10; для буквы Д — кодовое слово 01. Какова наименьшая возможная сумма длин кратчайших кодовых слов для букв В и Г?

*Примечание.* Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Ответ: _____.

**6** На вход алгоритма подаётся натуральное число N. Алгоритм строит по нему новое число R следующим образом.

1) Строится двоичная запись числа N.

2) К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 11100 преобразуется в запись 111001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

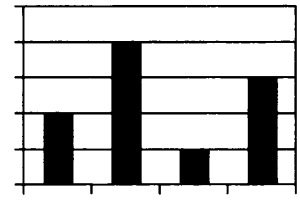
Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R.

Укажите минимальное число R, которое превышает 72 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Ответ: _____.

**7** Дан фрагмент электронной таблицы:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | 3 | | 7 | |
| 2 | =(B1-D2)/5 | =C1-A1 | =A1-2 | =C2*3 |



Какое число должно быть записано в ячейке B1, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?

Ответ: _____.

**8** Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

**Бейсик**

```
DIM k, s AS INTEGER
k = 100
s = 0
WHILE s + k < 180
  s = s + 25
  k = k - 10
WEND
PRINT s
```

**Паскаль**

```
var k, s : integer;
begin
  k := 100;
  s := 0;
  while s + k < 180 do
  begin
    s := s + 25;
    k := k - 10
  end;
  write(s)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
   int k, s;
   k = 100;
   s = 0;
   while (s + k < 180) {
     s = s + 25;
     k = k - 10;
   }
   cout << s << endl;
   return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел k, s
  k := 100
  s := 0
  нц пока s + k < 180
    s := s + 25
    k := k - 10
  кц
  вывод s
кон
```

**Python**

```
k = 100
s = 0
while s + k < 180:
    s = s + 25
    k = k - 10
print(s)
```

Ответ: _____.

**9** Документ объёмом 3 Гбайт можно передать с одного компьютера на другой двумя способами:

А. Сжать архиватором, передать по каналу связи, распаковать.

Б. Передать по каналу связи без использования архиватора.

Какой способ быстрее и на сколько, если:

- средняя скорость передачи данных по каналу связи составляет $2^{27}$ бит в секунду;
- объём сжатого архиватором документа равен 50% исходного;
- время, требуемое на сжатие документа, — 100 секунд, на распаковку — 20 секунд?

В ответе напишите букву А, если быстрее способ А, или Б, если быстрее способ Б. Сразу после буквы напишите число, обозначающее, на сколько секунд один способ быстрее другого.

Так, например, если способ Б быстрее способа А на 23 секунды, в ответе нужно написать Б23.

Единиц измерения «секунд», «сек.», «с.» к ответу добавлять не нужно.

Ответ: _____.

**10** Вася составляет 4-буквенные слова, в которых есть только буквы И, В, А, Н, причём буква А используется в каждом слове ровно 1 раз или не встречается вовсе. Каждая из других допустимых букв может встречаться в слове любое количество раз или не встречаться совсем. Словом считается любая допустимая последовательность букв, не обязательно осмысленная. Сколько существует таких слов, которые может написать Вася?

Ответ: _____.

**11** Даны рекурсивные алгоритмы F и G. Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(20)?

**Бейсик**

```
SUB F(n)
  IF n > 0 THEN
    G(n - 1)
  END IF
END SUB
SUB G(n)
  PRINT "*"
  IF n > 1 THEN
    F(n - 2)
  END IF
END SUB
```

**Паскаль**

```
procedure G(n: integer); forward;
procedure F(n: integer);
begin
  if n > 0 then
    G(n - 1)
end;
procedure G(n: integer);
begin
  write('*');
  if n > 1 then
    F(n - 2)
end;
```

**C++**

```
void F(int n);
void G(int n);
void F(int n) {
  if (n > 0)
    G(n - 1);
}
void G(int n) {
  std::cout << "*";
  if (n > 1)
    F(n - 2);
}
```

**Алгоритмический язык**

```
алг F(цел n)
нач
  если n > 0 то
    G(n - 1)
  все
кон
алг G(цел n)
нач
  вывод '*'
  если n > 1 то
    F(n - 2)
  все
кон
```

**Python**

```
def F(n):
    if n > 0:
        G(n - 1)
def G(n):
    print("*")
    if n > 1:
        F(n - 2)
```

Ответ: _____.

**12** В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. При этом адрес узла в сети (часть IP-адреса, задающего адрес самого узла в сети) не может состоять из одних нулей или из одних единиц. Сколько компьютеров может быть в сети с маской:

255.255.254.0?

Ответ: _____.

**13** Кабельная сеть проводит голосование среди зрителей о том, какой из 7-ми фильмов они хотели бы посмотреть вечером. Автоматизированная система голосования использует для кодирования номера каждого фильма минимально возможное и одинаковое количество бит. Кабельной сетью пользуются 5000 человек. В голосовании участвовало 2000 человек. Каков объём информации (в байтах), записанный автоматизированной системой голосования?

Ответ: _____.

**14** Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **сместиться на (a, b)**, где $a$, $b$ — целые числа. Эта команда перемещает Чертёжника из точки с координатами $(x, y)$ в точку с координатами $(x + a; y + b)$.

Например, если Чертёжник находится в точке с координатами (4, 2), то команда **сместиться на (2, −3)** переместит Чертёжника в точку (6, −1).

Цикл

**ПОВТОРИ число РАЗ**

**последовательность команд**

**КОНЕЦ ПОВТОРИ**

означает, что **последовательность команд** будет выполнена указанное число раз (число должно быть натуральным).

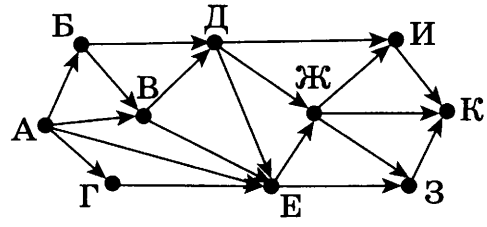
Чертёжнику был дан для исполнения следующий алгоритм (буквами n, a, b обозначены неизвестные числа, при этом n > 1):

**НАЧАЛО**
**сместиться на (12, −2)**
**ПОВТОРИ n РАЗ**
**сместиться на (a, b)**
**сместиться на (12, 8)**
**КОНЕЦ ПОВТОРИ**
**сместиться на (−47, −18)**
**КОНЕЦ**

Укажите наименьшее возможное значение числа $n$, для которого найдутся такие значения чисел $a$ и $b$, что после выполнения программы Чертёжник возвратится в исходную точку.

Ответ: _____.

**15** На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К, проходящих через город Е?




Ответ: _____.

**16** В системе счисления с основанием 4 записано четырёхзначное число. Известно, что все цифры числа разные. Какое самое маленькое число удовлетворяет этому условию? Ответ запишите в десятичной системе счисления.

Ответ: _____.

**17** В языке запросов к поисковому серверу для обозначения логической операции «ИЛИ» используется символ |, а для логической операции «И» — &. В таблице приведено количество страниц, которое находит поисковая система по каждому запросу.

| Запрос | Количество найденных страниц (в тысячах) |
|---|---|
| *Лампа* \| *Книга* \| *Блокнот* | 63 |
| *Книга* | 32 |
| *Лампа* | 40 |
| *Лампа & Блокнот* | 15 |
| *Лампа & Книга* | 21 |
| *Книга & Блокнот* | 0 |

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу *Блокнот* ?

Ответ: _____.

**18** Обозначим через ДЕЛ(n, m) утверждение «натуральное число n делится без остатка на натуральное число m».

Для какого наибольшего натурального числа A формула
¬ДЕЛ(x, A) → (ДЕЛ(x,6) → ¬ДЕЛ(x,8)) тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Ответ: _____.

**19** В программе используется целочисленный массив A с индексами от 0 до 9. Значения элементов массива равны 3; 5; 1; 7; 10; 6; 9; 11; 7; 8 соответственно, т.е. A[0] = 3; A[1] = 5 и т.д.

Определите значение переменной *k* после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

**Бейсик**

```
k = 0
FOR i = 0 TO 9
  IF A(i) MOD 2 = 0 AND k=0 THEN
    k = A(i)
  END IF
NEXT i
```

**Паскаль**

```
k := 0;
for i := 0 to 9 do
  if (A[i] mod 2 = 0) and (k=0) then
    k := A[i];
```

**C++**

```
k = 0;
for (i = 0 ; i <= 9 ; i++)
  if (A[i] % 2 == 0 && k == 0)
    k = A[i];
```

**Алгоритмический язык**

```
k := 0
нц для i от 0 до 9
  если mod(A[i],2) = 0 и k = 0 то
      k := A[i]
  все
кц
```

**Python**

```
k = 0
for i in range(10):
    if A[i] % 2 == 0 and k == 0:
        k = A[i]
```

Ответ: _____.

**20** Ниже на пяти языках записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *M*. Известно, что $x > 50$. Укажите наименьшее из таких чисел *x* (больших 50), при вводе которых алгоритм печатает 14.

**Бейсик**

```
DIM X, L, M AS INTEGER
INPUT X
M = 42
L = X
IF L MOD 2 = 0 THEN
  M = 28
END IF
WHILE L <> M
  IF L > M THEN
    L = L - M
  ELSE
    M = M - L
  END IF
WEND
PRINT M
```

**Паскаль**

```
var x, L, M: integer;
begin
  readln(x);
  M := 42;
  L := x;
  if L mod 2 = 0 then
    M := 28;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  write(M)
end.
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
  int x, L, M;
  cin >> x;
  M = 42;
  L = x;
  if (L % 2 == 0)
    M = 28;
  while (L != M)
    if(L > M)
      L = L - M;
    else
      M = M - L;
  cout << M << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел x, L, M
  ввод x
  M := 42
  L := x
  если mod(L,2)=0 то
    M := 28
  все
  нц пока L <> M
    если L > M то
      L := L - M
    иначе
      M := M - L
    все
  кц
  вывод M
кон
```

**Python**

```
x = int(input())
M = 42; L = x
if L % 2 == 0:
    M = 28
while L != M:
    if L > M:
        L = L - M
    else:
        M = M - L
print(M)
```

**Ответ:** _____.

**21** Ниже на пяти языках представлен алгоритм.

**Напишите в ответе число различных значений входной переменной k, при которых алгоритм выдаёт тот же ответ, что и при входном значении k = 60. Значение k = 60 также включается в подсчет различных значений k.**

**Бейсик**

```
DIM K, I AS INTEGER
INPUT K
I = 0
WHILE F(I) < K
  I = I + 1
WEND
IF F(I)+F(I-1) < 2*K THEN
  PRINT I
ELSE
  PRINT I-1
END IF

FUNCTION F (N)
  F = N * N - 10
END FUNCTION
```

**Паскаль**

```
var k, i : longint;
function f(n: longint):longint;
begin
  f := n * n - 10
end;
begin
  readln(k);
  i := 0;
  while f(i) < k do
    i := i + 1;
  if f(i)+f(i-1) < 2*k then
    writeln(i)
  else
    writeln(i-1)
end.
```

**C++**

```
#include <iostream>
using namespace std;
long f(long n) {
  return n * n - 10;
}
int main() {
  long k, i;
  cin >> k;
  i = 0;
  while (f(i) < k)
    i++;
  if (f(i) + f(i-1) < 2*k)
    cout << i << endl;
  else
    cout << i - 1 << endl;
  return 0;
}
```

**Алгоритмический язык**

```
алг
нач
 цел k, i
 ввод k
 i := 0
 нц пока F(i) < k
   i := i + 1
 кц
 если F(i)+F(i-1) < 2 * k то
   вывод i
 иначе
   вывод i - 1
 все
кон

алг цел F(цел n)
нач
  знач := n * n - 10
кон
```

**Python**

```
def f(n):
    return n * n - 10
k = int(input())
i = 0
while f(i) < k:
    i = i + 1
if f(i) + f(i - 1) < 2 * k:
    print(i)
else:
    print(i - 1)
```

Ответ: _____.

**22** Исполнитель Увеличитель преобразует число на экране.

У исполнителя есть три команды, которым присвоены номера:

**1. Прибавить 2**

**2. Умножить на 2**

**3. Умножить на 3**

Первая команда увеличивает число на экране на 3, вторая умножает его на 2.

Программа для исполнителя Увеличитель — это последовательность команд.

Сколько существует программ, для которых при исходном числе **1** результатом является число **58** и при этом траектория вычислений содержит число **18** и не содержит число **22**?

Траектория вычислений программы — это последовательность результатов выполнения всех команд программы. Например, для программы **123** при исходном числе 7 траектория будет состоять из чисел 8, 16, 48.

Ответ: _____.

**23** Сколько существует различных наборов значений логических переменных $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, $y_1$, $y_2$, $y_3$, $y_4$, $y_5$, которые удовлетворяют всем перечисленным ниже условиям?

$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge (\neg x_4 \vee x_5) = 1$

$(y_1 \to y_2) \wedge (y_2 \to y_3) \wedge (y_3 \to y_4) \wedge (y_4 \to y_5) = 1$

$(\neg y_1 \vee x_1) \wedge (\neg y_2 \vee x_2) \wedge (\neg y_3 \vee x_3) \wedge (\neg y_4 \vee x_4) \wedge (\neg y_5 \vee x_5) = 1$

В ответе **не нужно** перечислять все различные наборы значений $x_1$, $x_2$, $x_3$, $x_4$, $x_5$, $y_1$, $y_2$, $y_3$, $y_4$, $y_5$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

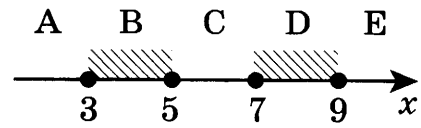
*Не забудьте перенести все ответы в бланк ответов № 1 в соответствии с инструкцией по выполнению работы.*

## Часть 2

*Для записи ответов на задания этой части (24–27) используйте БЛАНК ОТВЕТОВ № 2. Запишите сначала номер задания (24, 25 и т. д.), а затем полное решение. Ответы записывайте чётко и разборчиво.*

**24**

Требовалось написать программу, при выполнении которой с клавиатуры считывается координата точки на прямой (x — действительное число) и определяется принадлежность этой точки одному из выделенных отрезков B и D (включая границы). Программист торопился и написал программу неправильно.




**Паскаль**

```
var x: real;
begin
 readln(x);
 if x >= 3 then
  if x <= 9 then
   if x >= 7 then
   write('принадлежит')
 else
write('не принадлежит')
end.
```

**Бейсик**

```
INPUT x
IF x >= 3 THEN
 IF x <= 9 THEN
  IF x >= 7 THEN
   PRINT "принадлежит"
  ELSE
 PRINT "не принадлежит"
  ENDIF
 ENDIF
END
```

**C++**

```
#include <iostream>
using namespace std;
int main() {
 float x;
 cin >> x;
 if (x >= 3)
   if (x <= 9)
     if (x >= 7)
   cout << "принадлежит";
 else
 cout << "не принадлежит";
}
```

**Алгоритмический язык**

```
алг
нач
   вещ x
   ввод x
   если x >= 3 то
     если x <= 9 то
       если x >= 7 то
         вывод 'принадлежит'
       иначе
         вывод 'не принадлежит'
       все
     все
   все
кон
```

**Python**

```
x = float(input())
if x >= 3:
    if x <= 9:
        if x >= 7:
            print("принадлежит")
        else:
            print("не принадлежит")
```

Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (A, B, C, D и E). Границы (точки 3, 5, 7 и 9) принадлежат заштрихованным областям (B и D соответственно).

| Область | Условие 1 ($x$>=3) | Условие 2 ($x$<=9) | Условие 3 ($x$>=7) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| A | | | | | |
| B | | | | | |
| C | | | | | |
| D | | | | | |
| E | | | | | |

В столбцах условий укажите «да», если условие выполнится, «нет», если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа ведёт себя по-разному для разных значений, принадлежащих данной области. В столбце «Программа выведет» укажите, что программа выведет на экран. Если программа ничего не выводит, напишите «—» (прочерк). Если для разных значений, принадлежащих области, будут выведены разные тексты, напишите «не изв.». В последнем столбце укажите «Да» или «Нет».

2. Укажите, как нужно доработать программу, чтобы не было случаев её неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

**25** Дан целочисленный массив из 40 элементов. Элементы массива могут принимать значения от –10 000 до 10 000. Опишите на одном из языков программирования алгоритм, который проверяет, есть ли в массиве хотя бы один двузначный элемент. Программа должна вывести на экран «YES» или «NO».

Исходные данные объявлены так, как показано ниже. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать часть из них.

**Паскаль**

```
const N = 40;
var
    a: array [1..N] of integer;
    i, j, k : integer;
begin
   for i:=1 to N do
     readln(a[i]);
   ...
end.
```

**Бейсик**

```
N = 40
DIM A(N) AS INTEGER
DIM I, J, K AS INTEGER
FOR I = 1 TO N
  INPUT A(I)
NEXT I
...
END
```

**C++**

```
#include <iostream>
using namespace std;
const int N = 40;
int main() {
 long a[N];
 long i, j, k;
 for (i = 0; i < N; i++)
    cin >> a[i];
 ...
 return 0;
}
```

**Алгоритмический язык**

```
алг
нач
  цел N = 40
  целтаб a[1:N]
  цел i, j, k
  нц для i от 1 до N
    ввод a[i]
  кц
  ...
кон
```

**Python**

```
# допускается также использовать
# две целочисленные переменные j и k
a = []
n = 40
for i in range(0, n):
    a.append(int(input()))
...
```

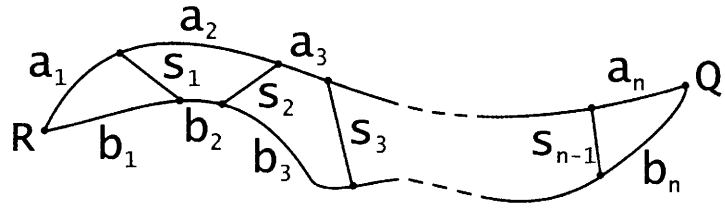
В качестве ответа вам необходимо привести фрагмент программы, который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6). В этом случае вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на языке Паскаль).

**26** Два игрока играют в следующую игру. На координатной плоскости в точке (1, 0) стоит фишка. Игроки ходят по очереди. Ход состоит в том, что игрок перемещает фишку из точки с координатами $(x, y)$ в одну из трёх точек: в точку с координатами $(x + 3, y)$, в точку $(x, y + 3)$ или в точку $(x + 3, y + 3)$. Выигрывает игрок, после хода которого расстояние по прямой от фишки до точки с координатами (0, 0) не меньше 13 единиц. Кто выигрывает при безошибочной игре обоих игроков — игрок, делающий первый ход, или игрок, делающий второй ход? Какими должны быть ходы выигрывающего игрока? Ответ обоснуйте.

**27** Из населённого пункта R в населённый пункт Q ведут две дороги. Назовём их A и B.

Дороги идут невдалеке друг от друга, не пересекаются. Периодически между ними встречаются соединительные дороги (связки), по которым можно переехать с дороги A на дорогу B и обратно.

Необходимо найти кратчайшее расстояние между населенными пунктами R и Q при условии, что можно ехать по любой дороге — A или B, и любое количество раз переезжать (если это нужно/короче) по дорогам-связкам с одной дороги на другую.




На вход программе подаётся: в первой строке — число дорог-связок N.

В каждой из последующих N строк — три целых неотрицательных числа:

расстояние от предыдущей «развилки» по дороге A, расстояние от предыдущей развилки по дороге B, длина дороги-связки (в последней строке длина дороги-связки равна нулю).

*Пример входных данных:*

```
4
3 7 3
12 2 4
3 10 5
7 6 0
```

*Пример выходных данных:*

```
22
```

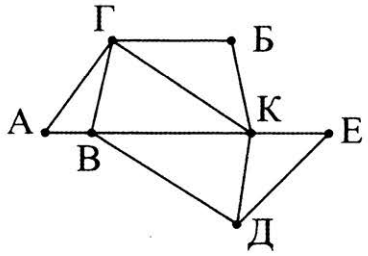
# ОТВЕТЫ

## Вариант 1

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 12 | yzx | 20 | 5 | CDBACED | BC | =$B2+D$2 | 127 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 256 | OOEO | 68 | CDBA | 32 | 4 | 30 | 7,14,28 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 8900 | 19 | 22 | 316 | 298 | 7 | 32 | |

Дополнение к ответу задания 3.

| | Б | Д | В | К | Е | А | Г |
|---|---|---|---|---|---|---|---|
| Б | | | | 60 | | | 45 |
| Д | | | 50 | 20 | 10 | | |
| В | | 50 | | 40 | | 30 | 25 |
| К | 60 | 20 | 40 | | 15 | | 55 |
| Е | | 10 | | 15 | | | |
| А | | | 30 | | | | 35 |
| Г | 45 | | 25 | 55 | | 35 | |




**Д-К**

**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | |
|---|---|
| Элементы ответа:<br>1) 2<br>2) 346 (или любое натуральное число, в котором первая цифра — единственная в числе нечётная цифра, большая 1, либо в котором первая цифра 1, а вторая цифра на 1 больше, чем сумма нечётных цифр в числе)<br>3) (Исправление программы на языке Паскаль)<br>В строке «`while N > 1 do`» должно быть «`while N > 0 do`», в строке «`writeln(d)`» должно быть «`writeln(sum)`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!** В задаче требовалось выполнить четыре действия (правильно ответить на вопросы 1 и 2 и исправить две ошибки). То есть:<br>1. Верно указать, что именно выведет программа при указанных в условии входных данных.<br>2. Указать число, при котором программа работает верно<br>3-4. Указать и верно исправить две ошибки:<br>а) Указать и верно исправить ошибку условия цикла.<br>б) Указать и верно исправить ошибку вывода на экран не той переменной.<br>Каждый из п. а) и б) считается выполненным, если:<br>i) правильно указана строка с ошибкой;<br>ii) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа. | |
| Правильно выполнены все пункты задания. Обе ошибки исправлены верно. Программа после исправлений для всех натуральных чисел N, не превосходящих $10^9$, верно определяет сумму нечётных цифр.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. | 3 |
| 2 балла за эту задачу выставляется в случае, если нельзя выставить 3 балла, и при этом выполняется один из двух случаев:<br>1) Правильно выполнены три действия из четырёх. Верное указание на ошибку при неверном исправлении при этом не засчитывается. При этом ни одна верная строка не указана как ошибочная.<br>2) Правильно выполнены все четыре действия. Не более одной верной строки указано как ошибочные. | 2 |
| Правильно выполнено только два действия из четырёх. Не важно, сколько верных строк указаны как неверные. | 1 |
| Все пункты задания выполнены неверно или отсутствуют. | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
j := 1;
while (j < N) and (a[j] <= 0) do
  j := j + 1;
if a[j] <= 0 then
  writeln('нет таких')
else
begin
  for i := j to N do
    if (a[i] > 0) and
       (a[i] < a[j]) then
      j := i;
  writeln(a[j])
end;
```

Другой способ:

```
j := 0;
for i := 1 to N do
  if a[i] > 0 then
    if (j = 0) or
       (a[i] < a[j]) then
      j := i;
if j = 0 then
  writeln('нет таких')
else
  writeln(a[j])
```

**На языке Бейсик**

```
J = 1
WHILE J < N AND A(J) <= 0
  J = J + 1
ENDW
IF A(J) <= 0 THEN
  PRINT "нет таких"
ELSE
  FOR I = J TO N
    IF A(I) > 0 AND
       A(I) < A(J) THEN
      J = I
    END IF
  NEXT I
  PRINT A(J)
END IF
```

**На языке C++**

```
j = 0;
while(j < N-1 && a[j] <= 0)
  j++;
if(a[j] <= 0)
  cout << "нет таких";
else
{
  for(i = j ; i < N ; i++)
    if(a[i] > 0 && a[i] < a[j])
      j = i;
  cout << a[j];
}
```

**На алгоритмическом языке**

```
j := 1
нц пока j < N и a[j] <= 0
  j := j+1
кц
если a[j] <= 0 то
    вывод 'нет таких'
иначе
  нц для i от j до N
    если a[i] > 0 и a[i] < a[j]
      то
        j := i
    все
  кц
  вывод a[j]
все
```

**На языке Python**

```
j = 0
while j < n-1 and a[j] <= 0:
    j = j + 1
if a[j] <= 0:
    print("нет таких")
else:
    for i in range(j, n):
        if a[i]> 0 and a[i] < a[j]:
            j = i
    print(a[j])
```

**26.**

**Содержание верного ответа и указания к оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

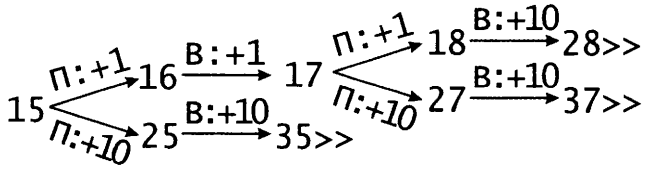
1а. При S=18...27. Во всех этих случаях Петя должен добавить в кучу 10 камней и выиграть. При значениях S<18 невозможно одним ходом (+1 или +10) получить 28 или больше камней.

1б. При S=17. Куда бы ни пошел Петя (17+1=18 или 17+10=27), Ваня добавит в кучу 10 камней и выиграет (18+10=28, 27+10=37).

2. S=16 или S=7. В обоих случаях Петя может получить в кучке 17 камней (16+1=17, 7+10=17). При любом ответном ходе Вани (17+1=18 или 17+10=27), Петя должен добавить в кучу 10 камней и выиграть (18+10=28, 27+10=37).

3. S=15. Если Петя добавит 10 камней в кучу (15+10=25), Ваня тоже должен добавить 10 камней в кучу и выиграть (25+10=35). Если Петя добавит 1 камень в кучу (15+1=16), Ваня также должен добавить 1 камень в кучу (16+1=17).

В ответ на любой следующий ход Пети (17+1=18 или 17+10=27), Ваня должен добавить 10 камней в кучу и выиграть (18+10=28, 27+10=37). Рассмотрим дерево игры:




В этом дереве в каждой позиции, где должен ходить Петя, разобраны все возможные ходы, а для позиций, где должен ходить Ваня — только ход, соответствующий стратегии, которую выбрал Ваня.

**27.**

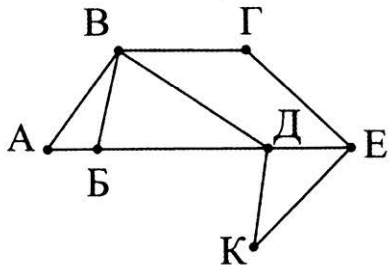
```
var
    i, j, x, max, max1, N : integer;
    a : array [1..5] of integer; // предыдущие введенные 5 элементов
begin
 readln(N);
 for i := 1 to 5 do
   readln(a[i]); // запоминаем первые 5 элементов последовательности
 max := 0; // начальное значение максимальной искомой суммы
 max1 := a[1]; // начальное значение максимального элемента, отстоящего
               // от текущего элемента не менее чем на 5
 for i := 6 to N do
 begin
   readln(x); // считываем очередной элемент
   if a[1] > max1 then // обновляем, при необходимости, текущий max1
     max1 := a[1];
   if x + max1 > max then // если сумма текущего элемента и отстоящего
                          // от него на 5 элементов ранее максимума больше max
     max := x + max1; // считаем ее новым максимумом
   for j := 1 to 4 do // сдвигаем массив 5-ти последних элементов на 1
     a[j] := a[j+1];
   a[5] := x          // текущий элемент запоминаем в его "хвосте"
 end;
 writeln(max)
end.
```

## Вариант 2

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 3 | yxz | 8 | 3 | 101 | DBx | 8 | 33 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 80 | 4 | 15 | FCDA | 400 | 27 | 41 | 5,13,21 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 1500 | 12 | 5 | 544 | 10 | 47 | 108 | |

**Дополнение к ответу задания 3.**

| | Д | К | А | В | Е | Б | Г |
|---|---|---|---|---|---|---|---|
| Д | | 25 | | 13 | 16 | 10 | |
| К | 25 | | | | 7 | | |
| А | | | | 21 | | 14 | |
| В | 13 | | 21 | | | 11 | 5 |
| Е | 16 | 7 | | | | | 8 |
| Б | 10 | | 14 | 11 | | | |
| Г | | | | 5 | 8 | | |




**Г-Е**

**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | |
|---|---|
| Элементы ответа:<br>1) «NO»<br>2) 1 2 3 4 (или любая последовательность 4-х целых чисел, у которой сумма чётных чисел больше нуля)<br>3) (Исправление программы на языке Паскаль)<br>В строке «`count := 1;`» должно быть «`count := 0;`».<br>В строке «`if sum > 0 then`» должно быть «`if count > 0 then`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!** В задаче требовалось выполнить четыре действия:<br>1. Указать, что выведет программа для указанных входных данных,<br>2. Привести пример входных данных, при которых программа работает верно,<br>3. Исправить одну ошибку в программе,<br>4. Исправить вторую ошибку в программе | |
| Правильно выполнены все пункты задания. Ошибки исправлены верно.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. Ни одна правильная строка не указана в качестве неправильной. | 3 |
| Правильно выполнены три действия из четырёх. Верное указание на ошибку при неверном исправлении при этом не засчитывается. Не более одной верной строки указано как неправильная. | 2 |
| Правильно выполнено два действия из четырёх. Не более одной верной строки указано в качестве неверной. | 1 |
| Правильно выполнено менее двух действий | 0 |
| **Максимальный балл** | 3 |

**25.**

| **На языке Паскаль** | **На языке Бейсик** |
|---|---|
| `sum := 0;`<br>`for i := 1 to n do`<br>`  if (a[i] < 300) and`<br>`  (a[i] mod 10 <> 5) then`<br>`  sum := sum + a[i];`<br>`for i := 1 to n do`<br>`  if (a[i] < 300) and`<br>`  (a[i] mod 10 <> 5) then`<br>`  a[i] := sum;`<br>`for i := 1 to n do`<br>`  writeln(a[i])` | `SUM = 0`<br>`FOR I = 1 TO N`<br>`  IF A(I) < 300 AND`<br>`  A(I) MOD 10 <> 5 THEN`<br>`  SUM = SUM + A(I)`<br>`  END IF`<br>`NEXT I`<br>`FOR I = 1 TO N`<br>`  IF A(I) < 300 AND`<br>`  A(I) MOD 10 <> 5 THEN`<br>`  A(I) = SUM`<br>`  END IF`<br>`  PRINT A(I)`<br>`NEXT I` |

**На языке C++**

```
sum = 0;
for(i = 0 ; i < N ; i++)
  if(a[i] < 300 && a[i] % 10 != 5)
    sum += a[i];
for(i = 0 ; i < N ; i++) {
  if(a[i] < 300 && a[i] % 10 != 5)
    a[i] = sum;
  cout << a[i] << endl;
}
```

**На алгоритмическом языке**

```
sum := 0
нц для i от 1 до N
  если a[i] < 300 и mod(a[i],10)<>5
    то
      sum := sum + a[i]
  все
кц
нц для i от 1 до N
  если a[i] < 300 и mod(a[i],10)<>5
    то
      a[i] := sum
  все
  вывод a[i], нс
кц
```

**На языке Python**

```
sum = 0
for i in range(0, n):
    if a[i] < 300 and a[i] % 10 != 5:
        sum = sum + a[i]
for i in range(0, n):
    if a[i] < 300 and a[i] % 10 != 5:
        a[i] = sum
    print(a[i])
```

**26.**

**Содержание верного ответа и указания к оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

1а. При S=15...29. Во всех этих случаях Петя должен увеличить количество камней в куче в два раза и выиграть. При значениях S<15 невозможно одним ходом (+2, +3 или *2) получить 30 или больше камней.

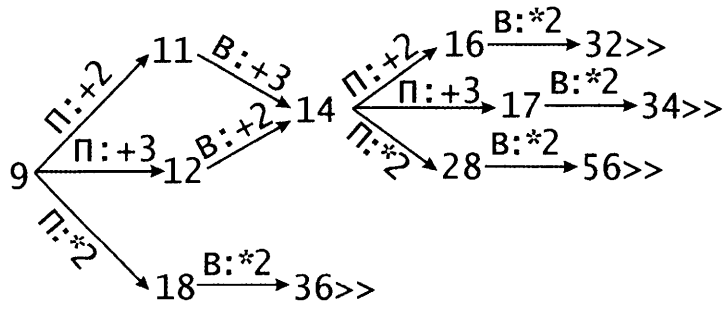
1б. При S=13 или S=14. Куда бы ни пошел Петя (13+2=15, 13+3=16, 13*2=26, 14+2=16, 14+3=17 или 14*2=28), Ваня удвоит число камней в куче и выиграет (15*2=30, 16*2=32, 17*2=34, 26*2=52 или 28*2=56).

2. Верный ответ: любые 2 значения среди S=7, S=10, S=11 или S=12. Например, S=11 или S=12. В обоих случаях Петя должен получить в кучке 14 камней (11+3=14, 12+2=14). При любом ответном ходе Вани (14+2=16, 14+3=17 или 14*2=28), Петя должен удвоить число камней в куче и выиграть (16*2=32, 17*2=34 или 28*2=56).

3. S=9 (также верный ответ S=8). Если Петя добавит в кучу 2 или 3 камня (9+2=11 или 9+3=12), Ваня должен получить в куче 14 камней (11+3=14, 12+2=14) и в ответ на любой ход Пети (14+2=16, 14+3=17 или 14*2=28), Ваня должен удвоить число камней в куче и выиграть (16*2=32, 17*2=34 или 28*2=56). Если Петя удвоит число камней в куче (9*2=18), Ваня также должен удвоить число камней в куче и выиграть (18*2=36).

Рассмотрим дерево игры (см. рис.).

Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.




**27.**

```
var ch,cmax:char;
  num:array['A'..'Z']of integer; {массив количества букв алфавита}
  k:integer;
```

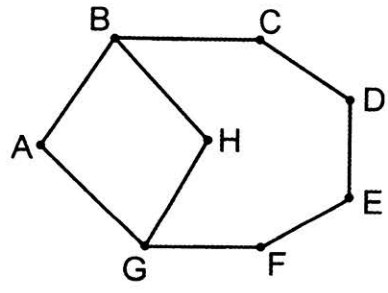
```
begin
 for ch:='A' to 'Z' do
   num[ch]:=0;    {обнуляем массив}
 read(ch);
 while ch<>'#' do
 begin
   {если текущий символ - буква}
   if (upcase(ch)>='A') and (upcase(ch)<='Z') then
     inc(num[upcase(ch)]);
   read(ch)
 end;
 k:=1;
 cmax:='A';
 for ch:='B' to 'Z' do
   if num[ch]>num[cmax] then
   begin
     cmax:=ch;
     k:=1
   end
   else
     if num[ch]=num[cmax] then
       k:=k+1;
 if k=1 then
   writeln(cmax)
 else
   for ch:='A' to 'Z' do
     if num[ch]=num[cmax] then
       write(ch,' ');
 writeln(k)
end.
```

## Вариант 3

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 2 | wyzx | 58 | 1 | C1 | 157 | 3 | 17 |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| 1 | 5 | 13 | 23 | 600 | 8877 | 25 | 99 |
| **17** | **18** | **19** | **20** | **21** | **22** | **23** | |
| 11 | 65 | 5 | 997 | 134 | 25 | 32 | |

**Дополнение к ответу задания 3.**

| | B | F | G | C | E | A | H | D |
|---|---|---|---|---|---|---|---|---|
| **B** | | | | * | | * | * | |
| **F** | | | * | | * | | | |
| **G** | | * | | | | * | * | |
| **C** | * | | | | | | | * |
| **E** | | * | | | | | | * |
| **A** | * | | * | | | | | |
| **H** | * | | * | | | | | |
| **D** | | | | * | * | | | |




**D-E**

**24.**

| Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла) | |
|---|---|
| Элементы ответа:<br>1) 0<br>2) 0 0 0 0 (или любая последовательность 4-х целых чисел, в которой есть хотя бы один ноль, а другие числа отрицательные)<br>3) (Исправление программы на языке Паскаль)<br>В строке «`max := 0;`» должно быть «`max := -1001`» (или любое число <=1000).<br>В строке «`a := max`» должно быть «`max := a`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!** В задаче требовалось выполнить четыре действия:<br>1. Указать, что выведет программа для указанных входных данных,<br>2. Привести пример входных данных, при которых программа работает верно,<br>3. Исправить одну ошибку в программе,<br>4. Исправить вторую ошибку в программе | |
| Правильно выполнены все пункты задания. Ошибки исправлены верно.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. Ни одна правильная строка не указана в качестве неправильной. | 3 |
| Правильно выполнены три действия из четырёх. Верное указание на ошибку при неверном исправлении при этом не засчитывается. Не более одной верной строки указано как неправильная. | 2 |
| Правильно выполнено два действия из четырёх. Не более одной верной строки указано в качестве неверной. | 1 |
| Правильно выполнено менее двух действий | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
m := -20001;
for i := 1 to N-1 do
  if ((a[i] + a[i+1]) mod 3 <> 0)
     and (a[i] + a[i+1] > m) then
    m := a[i] + a[i+1];
if m = -20001 then
   m := 0;
writeln(m)
```

**На языке Бейсик**

```
M = -20001
FOR I = 1 TO N-1
IF ((A(I)+A(I+1)) MOD 3 <> 0) AND
   (A(I) + A(I+1) > M) THEN
  M = A(I) + A(I+1)
END IF
NEXT I
IF M = 20001 THEN
   M = 0
END IF
PRINT M
```

**На языке C++**

```
m = -20001;
for(i = 0 ; i < N-1 ; i++)
  if( (a[i] + a[i+1]) % 3 != 0 &&
       a[i] + a[i+1] > m )
    m = a[i] + a[i+1];
if (m == -20001)
   m = 0;
cout << m;
```

**На алгоритмическом языке**

```
m := -20001
нц для i от 1 до N-1
  если mod(a[i]+a[i+1],3) <> 0 и
       a[i]+a[i+1] > m то
    m := a[i]+a[i+1]
  все
кц
если m = -20001 то
   m := 0
все
вывод m
```

**На языке Python**

```
m = -20001
for i in range(0, n-1):
    if (a[i] + a[i+1]) % 3 != 0 and a[i] + a[i+1] > m:
        m = a[i] + a[i+1]
if m == -20001:
   m = 0
print(m)
```

**26.**

**Содержание верного ответа и указания к оцениванию**

**(допускаются иные формулировки ответа, не искажающие его смысла)**

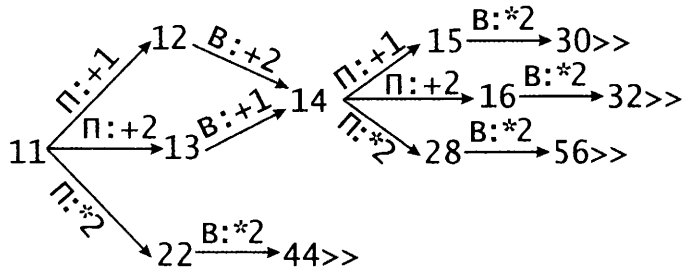
1а. При S=15...28. Во всех этих случаях Петя должен увеличить количество камней в куче в два раза и выиграть. При значениях S<15 невозможно одним ходом (+1, +2 или *2) получить 29 или больше камней.

1б. При S=14. Куда бы ни пошел Петя (14+1=15, 14+2=16 или 14*2=28), Ваня удвоит число камней в куче и выиграет (15*2=30, 16*2=32 или 28*2=56).

2. S=7, S=12 или S=13. Во всех случаях Петя должен получить в кучке 14 камней (7*2=14, 12+2=14, 13+1=14). При любом ответном ходе Вани (14+1=15, 14+2=16 или 14*2=28), Петя должен удвоить число камней в куче и выиграть (15*2=30, 16*2=32 или 28*2=56).

3. S=11. Если Петя добавит в кучу 1 или 2 камня (11+1=12 или 11+2=13), Ваня должен получить в куче 14 камней (12+2=14, 13+1=14) и в ответ на любой ход Пети (14+1=15, 14+2=16 или 14*2=28), Ваня должен удвоить число камней в куче и выиграть (15*2=30, 16*2=32 или 28*2=56). Если Петя удвоит число камней в куче (11*2=22), Ваня также должен удвоить число камней в куче и выиграть (22*2=44).

Рассмотрим дерево игры:




Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.

**27.**

```
var S,Smax,Smax2:string[52];
    ch:char;
    i,N,sh,ball,max,nmax,max2,nmax2:integer;
begin
 max:=-1;  Smax:='';  nmax:=0;
 max2:=-1;
 readln(N); {считали количество строк}
 for i:=1 to N do {перебираем все входные строки}
 begin
   s:='';
   repeat
     read(ch);
     s:=s+ch
   until ch=' ';{считана фамилия и запомнена в переменной s}
   repeat
     read(ch);
     s:=s+ch
   until ch=' ';{считано имя и добавлена к переменной s}
   readln(sh,ball); {считали номер школы и балл ученика}
   if sh=50 then    {обрабатываем только учеников 50-й школы}
     if ball>max then {текущий балл - лучший}
     begin
       max2:=max;     Smax2:=Smax;   nmax2:=nmax;
       max :=ball;    Smax :=s;      nmax :=1
     end
```

```
      else
        if ball=max then {текущий балл - такой же, как лучший}
        begin
          nmax:=nmax+1;
          max2:=max;  Smax2:=S
        end
        else
          if ball>max2 then {текущий балл - лучше второго}
          begin
            max2:=ball;  Smax2:=S;
            nmax2:=1
          end
          else
            if ball=max2 then {текущий балл такой же, как второй}
              nmax2:=nmax2+1
  end;
  if (nmax=2) or (nmax=1) and (nmax2=1) then {два лучших ученика}
  begin
    writeln(Smax);
    writeln(Smax2)
  end
  else
    if (nmax=1) and (nmax2>1) then {один лучший ученик}
      writeln(Smax)
    else
      writeln(nmax) {лучших учеников больше двух}
end.
```

## Вариант 4

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 6 | 45 | 14 | 134 | ABCA | 960 | 6 | 20 |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| A39 | 50 | 252432 | 240 | 160 | 5545 | 32 | 274 |
| **17** | **18** | **19** | **20** | **21** | **22** | **23** | |
| 13 | 219 | 3 | 74 | 63 | 70 | 364 | |

**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | **Баллы** |
|---|---|

Элементы ответа:

1)

| **Область** | **Условие 1 ($y<=x$)** | **Условие 2 ($y<=-x$)** | **Условие 3 ($y>=x*x-2$)** | **Программа выведет** | **Область обрабатывается верно** |
|---|---|---|---|---|---|
| **A** | нет | — | — | — | нет |
| **B** | нет | — | — | — | нет |
| **C** | нет | — | — | — | нет |
| **D** | нет | — | — | — | нет |
| **E** | да | нет | — | — | нет |
| **F** | да | да | нет | не принадлежит | да |
| **G** | да | да | да | принадлежит | да |
| **H** | нет | — | — | — | нет |
| **I** | да | нет | — | — | нет |

2) Возможная доработка (Паскаль):

```
if (x<=0) and (y>=x*x-2) and (y<=-x) or
   (x>=0) and (y>=x*x-2) and (y<=x) then
   write('принадлежит')
 else
   write('не принадлежит')
```

Возможны и другие способы доработки.

*Пример*:

```
if (y>=x*x-2) and ((y<=-x) or (y<=x)) then
   write('принадлежит')
 else
   write('не принадлежит')
```

| Указания по оцениванию | |
|---|---|
| **Обратите внимание!**<br>В задаче требовалось выполнить три действия: указать для каждой области, как будет работать программа, что она выведет на экран и правильно ли это (в виде таблицы), и исправить две ошибки.<br>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.<br>1. Верное заполнение предложенной таблицы.<br>2. Исправление неправильного использования условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.<br>В сложных случаях это действие считается выполненным, если программа выдает одно из двух сообщений «принадлежит» или «не принадлежит» для любых чисел x и y, при этом программа не стала работать хуже, чем раньше, то есть для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.<br>3. Исправление ошибки, из-за которой приведённым трём ограничениям не удовлетворяют точки плоскости, у которых $y \geqslant x^2-2$ и $y>x$ и $y \leqslant -x$ (область H), а также те, у которых $y \geqslant x^2-2$ и $y>-x$ и $y \leqslant x$ (область I). Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области её части.<br>В сложных случаях это действие считается выполненным, если верно определена закрашенная область, то есть программа выводит сообщение «принадлежит» для всех точек закрашенной области и только для них, для точек вне закрашенной области программа выводит «не принадлежит» или не выводит ничего. | |
| Правильно выполнены оба пункта задания.<br>Исправлены две ошибки.<br>Программа для всех пар чисел (x, y) верно определяет принадлежность точки закрашенной области.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. | 3 |
| 1. Правильно выполнены два действия из трёх (исправлены обе ошибки, но в первом пункте задания не приведена таблица (либо таблица содержит ошибки хотя бы в одной строке), либо приведена таблица (которая содержит ошибки не более чем в одной строке), но исправлена только одна ошибка программы).<br>При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо «$y \geqslant -x$» используется «$y>-x$».<br>2. Или выполнены все три действия, но при этом в логическом выражении неверно учтены приоритеты логических операций (не расставлены или неправильно расставлены скобки в выражениях). | 2 |
| Правильно выполнено только одно действие из трёх, то есть, либо только приведена таблица, которая содержит ошибки в не более чем двух строках, либо таблица не приведена (или приведена и содержит ошибки более чем в двух строках), но исправлена одна ошибка программы.<br>При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот). | 1 |
| Все пункты задания выполнены неверно (таблица анализа правильности алгоритма не приведена, либо содержит ошибки более чем в двух строках, программа не приведена, либо ни одна из двух ошибок не исправлена). | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
p := 1;
for i:=1 to N do
  if (a[i] mod 2 <> 0) and
     (a[i] > 0) then
    p := p * a[i];
writeln(p)
```

**На языке Бейсик**

```
P = 1
FOR I = 1 TO N
  IF A(I) MOD 2 <> 0 AND
     A(I) > 0 THEN
    P := P * A(I)
  END IF
NEXT I
PRINT P
```

**На языке C++**

```
p = 1;
for(i = 0 ; i < N ; i++)
  if( a[i] % 2 != 0 &&
      a[i] > 0 )
    p *= a[i];
cout << p;
```

**На алгоритмическом языке**

```
p := 1
нц для i от 1 до N
  если mod(a[i],2) <> 0 и a[i] > 0
  то
    p := p * a[i]
  все
кц
вывод p
```

**На языке Python**

```
p = 1
for i in range(0, n):
    if a[i] % 2 != 0 and a[i] > 0:
        p = p * a[i]
print(p)
```

**26.**

**Содержание верного ответа и указания к оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

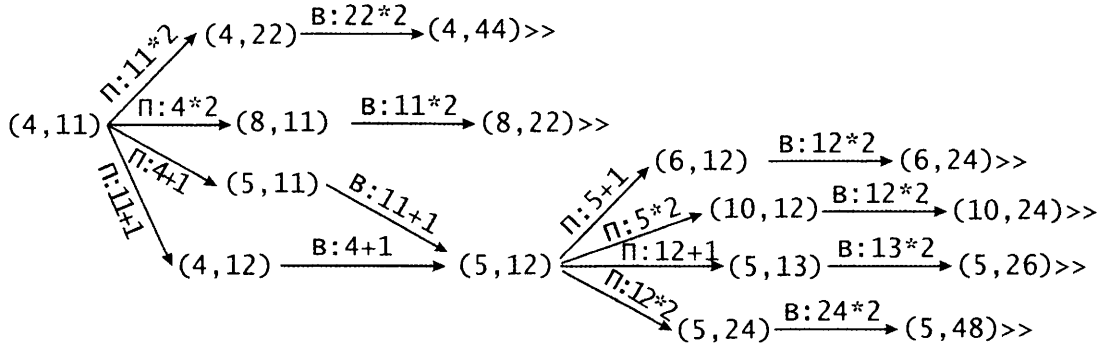
**1а)** S=13...25. Петя должен удвоить число камней во второй куче.

**1б)** 0 (нет таких S). Единственный подходящий вариант (S=12) не подходит, т.к. при первом ходе Пети «+1 к первой куче» получается ситуация (5,12), из которой нельзя выиграть одним ходом. Числа, меньшие 12, тоже не подходят (при S=11 добавление 1 камня во вторую кучу приводит к (4, 12), из которой не выиграть). Аналогично, можно такое показать для каждого меньшего S.

**2)** S=12. Первым ходом Петя должен добавить 1 камень в первую кучу (получится (5,12)). При любом ответном ходе Вани Петя должен удвоить вторую кучу и выиграть.

**3)** S=11. Если Петя добавит 1 камень в какую-нибудь кучку, Ваня должен добавить 1 камень в другую кучку. Получится ситуация (5,12). При любом ответном ходе Пети Ваня должен удвоить вторую кучу и выиграть. Если Петя удвоит какую-нибудь кучку, Ваня должен удвоить вторую кучу и выиграть.

Рассмотрим дерево игры:




В дереве рассмотрены все возможные ходы Пети и ответные ходы Вани, соответствующие выигрышной стратегии.

Знаком «>>» обозначены позиции, в которых партия заканчивается.

**27.**

```
var nmax,max:array[1..99] of integer;
    ch:char;
    i,N,sh,ball,k:integer;
begin
 for i:=1 to 99 do {обнуляем массивы}
 begin
   nmax[i]:=0;
   max[i]:=-1
 end;
 readln(N); {считали количество строк}
 for i:=1 to N do {перебираем все входные строки}
 begin
   repeat
     read(ch)
   until ch=' ';{считана фамилия}
   repeat
     read(ch)
   until ch=' ';{считано имя}
   readln(sh,ball); {считали номер школы и балл ученика}
   if ball>max[sh] then {сравниваем текущий балл с лучшим баллом по школе}
   begin
     max[sh]:=ball; {меняем лучший балл по школе}
     nmax[sh]:=1    {число людей в школе с таким баллом устанавливаем =1}
   end
   else
     if ball=max[sh] then {в школе есть еще такой лучший балл}
       nmax[sh]:=nmax[sh]+1; {увеличиваем число людей в школе с таким баллом}
 end;
 k:=0;
 for i:=1 to 99 do
   if nmax[i]>2 then {отбираем только школы, у которых лучший балл}
   begin             {набрало более 2-х учеников}
     k:=k+1;         {считаем количество таких школ}
     ball:=max[i];   {запоминаем балл в какой-нибудь из них}
     write(i,' ')    {выводим номер такой школы}
   end;
 if k=0 then
   writeln('Нет таких школ')
 else
 begin
   writeln;
   if k=1 then
     writeln('Наибольший балл = ',ball)
 end
end.
```

## Вариант 5

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 64 | 34 | 13 | 3 | 6 | 66 | 3 | 75 |
| **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| Б10 | 1458 | 35 | 62 | 300 | 3 | 21 | 333 |
| **17** | **18** | **19** | **20** | **21** | **22** | **23** | |
| 3 | 36 | 39 | 210 | 12 | 45 | 16 | |

**24.**

**Содержание верного ответа и указания по оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

Элементы ответа:

1.

| Область | Условие 1 ($x>1$) | Условие 2 ($x>=7$) | Условие 3 ($x<=13$) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| **A** | нет | — | — | — | нет |
| **B** | нет | — | — | — | нет |
| **C** | да | нет | — | — | нет |
| **D** | да | да | да | принадлежит | да |
| **E** | да | да | нет | не принадлежит | да |

2. Возможная доработка (Паскаль):

```
if (x>=-5) and (x<=1) or (x>=7) and (x<=13) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

Например:

```
if x>=-5 then
  if x<=1 then
    write('принадлежит')
  else
    if x>=7 then
      if x<=13 then
        write('принадлежит')
      else
        write('не принадлежит')
    else
      write('не принадлежит')
else
  write('не принадлежит')
```

| Указания по оцениванию | Баллы |
|---|---|
| **Обратите внимание!** В задаче требуется выполнить **три** действия.<br>1. Заполнить таблицу.<br>2. Исправить ошибку в условном операторе.<br>3. Исправить ошибку, связанную с неправильным набором условий.<br>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.<br>1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.<br>2. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.<br>В сложных случаях это действие считается выполненным, если программа выдает одно из двух сообщений: «принадлежит» или «не принадлежит» – для любых чисел $x$, при этом программа не стала работать хуже, чем раньше, т.е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ. | |
| 3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие $x \geqslant -5$). Кроме того, необходимо учесть, что области не соединены. Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции, либо использование сложной комбинации каскадных условий.<br>В сложных случаях это действие считается выполненным, если верно определены заштрихованные области, т.е. программа выводит сообщение «принадлежит» для всех точек закрашенных областей, и только для них, для точек вне заштрихованных областей программа выводит «не принадлежит» или не выводит ничего.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения | |

| | |
|---|---|
| Выполнены все три действия | 3 |
| Правильно выполнены два действия из трёх (исправлены обе ошибки, но в пункте 1 задания не приведена таблица (либо таблица содержит ошибки в двух и более строках), либо приведена таблица (которая содержит ошибки не более чем в одной строке), но исправлена только одна ошибка программы). При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо «$x \geqslant -5$» используется «$x > -5$» | 2 |
| Правильно выполнено только одно действие из трёх, т.е. либо только приведена таблица, которая содержит ошибки в не более чем одной строке, либо таблица не приведена (или приведена и содержит ошибки более чем в одной строке), но исправлена одна ошибка программы.<br>При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот) | 1 |
| Все пункты задания выполнены неверно (таблица анализа правильности алгоритма не приведена либо содержит ошибки в двух и более строках, программа не приведена либо ни одна из двух ошибок не исправлена) | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
k := 0;
j := 0;
for i:=1 to N do
  if a[i] > 0 then
    k := k + 1
  else
    if a[i] < 0 then
     j := j + 1;
if k > j then
  writeln('+')
else
  if k < j then
    writeln('-')
  else
    writeln('=')
```

**На языке Бейсик**

```
K = 0
J = 0
FOR I = 1 TO N
  IF A(I) > 0 THEN
    K = K + 1
  END IF
  IF A(I) < 0 THEN
    J = J + 1
  END IF
NEXT I
IF K = J THEN
  PRINT "="
ELSE
  IF K < J THEN
    PRINT "-"
  ELSE
    PRINT "+"
END IF
```

**На языке C++**

```
k = 0;
j = 0;
for(i = 0 ; i < N ; i++)
{
  if(a[i] > 0)
    k++;
  if(a[i] < 0)
    j++;
}
if(k > j)
  cout << "+";
else
  if(k < j)
    cout << "-";
  else
    cout << "=";
```

**На алгоритмическом языке**

```
k := 0
j := 0
нц для i от 1 до N
  если a[i] > 0 то
    k := k + 1
  все
  если a[i] < 0 то
    j := j + 1
  все
кц
если k = j то
    вывод '='
иначе
  если k < j то
    вывод '-'
  иначе
    вывод '+'
  все
все
```

**На языке Python**

```
k = 0; j = 0
for i in range(0, n):
    if a[i] > 0:
        k = k + 1
    if a[i] < 0:
        j = j + 1
if k > j:
    print("+")
else:
    if k < j:
        print("-")
    else:
        print("=")
```

**26.**

**Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)**

Выигрывает Ваня. Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны пары чисел, разделенные запятой. Эти числа соответствуют количеству камней на каждом этапе игры в первой и второй кучках соответственно.

| Стартовая позиция | I ход (Пети), все варианты хода | II ход (Вани), выигрышные ходы | III ход (Пети), все варианты хода | IV ход (Вани), выигрышные ходы |
|---|---|---|---|---|
| 4,3 | 12,3 | 36,3 | Выигрыш Вани | |
| | 6,3 | 6,9 | 18,9 | 54,9 |
| | | | 8,9 | 8,27 |
| | | | 6,27 | Выигрыш Вани |
| | | | 6,11 | 6,33 |
| | 4,9 | 6,9 | Те же варианты III–IV ходов | |
| | 4,6 | 12,6 | 36,6 | Выигрыш Вани |
| | | | 14,6 | 42,6 |
| | | | 12,18 | 36,18 |
| | | | 12,8 | 36,8 |

Таблица содержит все возможные варианты ходов первого игрока (Пети). Из неё видно, что при любом ходе первого игрока у второго (Вани) имеется ход, приводящий к победе.

**27.**

Решение, неэффективное по памяти и по времени (на 2 балла):

```
var
    i, j, k, N : integer;
    a : array [1..10000] of integer; // все элементы
begin
  readln(N);
  for i := 1 to N do
    readln(a[i]);
  k := 0;
  for i := 1 to n - 4 do
    for j := i + 4 to n do
      if a[i] * a[j] mod 6 = 0 then
        k := k + 1;
  writeln(k)
end.
```

**Решение, эффективное как по времени, так и по памяти:**

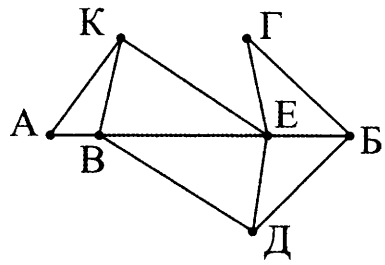
```
var
    i, j, k, k2, k3, k6, b, N : integer;
    a : array [1..3] of integer; // предыдущие введенные 3 элемента
begin
 readln(N);
 for i := 1 to 3 do
   readln(a[i]); // запоминаем первые 5 элементов последовательности
 k := 0; // искомое количество пар
 k2 := 0; // количество уже обработанных элементов, кратных 2, но не 3
 k3 := 0; // количество уже обработанных элементов, кратных 3, но не 2
 k6 := 0; // количество уже обработанных элементов, кратных 6
 for i := 4 to N do
 begin
   readln(b); // считываем очередной элемент
   // в зависимости от того, на что он кратен,
   // он может образовать пару с каким-то количеством
   // уже обработанных элементов
   if b mod 2 = 0 then
     if b mod 3 = 0 then // если кратен 6,
       k := k + (i-4)    // то со всеми обработанными
     else                // если кратен 2, но не 3,
       k := k + k3 + k6  // то с кратными 3, но не 2 и с кратными 6
   else
     if b mod 3 = 0 then // если кратен 3, но не 2
       k := k + k2 + k6  // то с кратными 2, но не 3 и с кратными 6
     else                // если не кратен 2 и 3,
       k := k + k6;      // то только с кратными 6
   // перед тем, как удалить элемент, хранимый в a[1],
   // обработаем его (посчитаем, к какой категории он принадлежит)
   if a[1] mod 2 = 0 then
     if a[1] mod 3 = 0 then
       k6 := k6 + 1
     else
       k2 := k2 + 1
   else
     if a[1] mod 3 = 0 then
       k3 := k3 + 1;
   for j := 1 to 2 do // сдвигаем массив 3-х последних элементов на 1
     a[j] := a[j+1];
   a[3] := b          // текущий элемент запоминаем в его "хвосте"
 end;
 writeln(k)
end.
```

## Вариант 6

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 10 | yxz | 35 | 4218 | CBBAE | AD | =$C1-A$3 | 41 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 512 | ДДБД | 64 | CBDA | 11 | 4 | 12 | 6,8,12,24 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 2200 | 13 | 5 | 58 | 94 | 14 | 15 | |

Дополнение к ответу задания 3.

| | Г | В | Е | А | Д | К | Б |
|---|---|---|---|---|---|---|---|
| Г | | | 65 | | | | 60 |
| В | | | 30 | 25 | 40 | 45 | |
| Е | 65 | 30 | | | 55 | 50 | 20 |
| А | | 25 | | | | 15 | |
| Д | | 40 | 55 | | | | 35 |
| К | | 45 | 50 | 15 | | | |
| Б | 60 | | 20 | | 35 | | |




**Б-Д**

**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | |
|---|---|
| Элементы ответа:<br>1) 2<br>2) 5 (или любое однозначное число или число, в котором все цифры одинаковые)<br>3) (Исправление программы на языке Паскаль)<br>В строке «`k := 10;`» должно быть «`k := 0;`»,<br>в строке «`if N mod 10 < k then`» должно быть «`if N mod 10 > k then`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!**<br>В задаче требовалось выполнить три действия.<br>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.<br>1. Верно указано, что именно выведет программа при указанных в условии входных данных.<br>2. Указано число, при котором программа работает верно<br>3. Указаны и верно исправлены две ошибки:<br>а) Указана и верно исправлена ошибка инициализации (не обязательно с упоминанием этого термина).<br>б) Указано на неверное условие выбора максимума, и оно исправлено на верное.<br>Каждый из п. а) и б) считается выполненным, если:<br>i) правильно указана строка с ошибкой;<br>ii) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа. | |
| Правильно выполнены все пункты задания. Обе ошибки исправлены верно.<br>Программа после исправлений для всех натуральных чисел N, не превосходящих $10^9$, верно определяет наибольшую цифру.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. | 3 |
| Правильно выполнены два действия из трёх.<br>Верное указание на ошибку при неверном исправлении при этом не засчитывается.<br>Допустимо, чтобы в пункте 3 была исправлена только одна ошибка. | 2 |
| Правильно выполнено только одно действие из трёх, то есть, либо только выполнен пункт 1, либо пункт 2, либо верно исправлена хотя бы одна ошибка путем её явного указания и исправления или в новом тексте программы | 1 |
| Все пункты задания выполнены неверно или отсутствуют. | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
j := 0;
for i := N downto 1 do
  if a[i] = x then
    j := i;
if j = 0 then
  writeln('нет таких')
else
  writeln(j);
```

Другой способ:

```
i := 1;
while (i < N) and (a[i] <> x) do
  i := i + 1;
if a[i] <> x then
  writeln('нет таких')
else
  writeln(i);
```

**На языке Бейсик**

```
I = 1
WHILE I<N AND A(I)<>X
  I = I + 1
ENDW
IF A(I)<>X THEN
  PRINT "нет таких"
ELSE
  PRINT I
END IF
```

**На языке C++**

```
i = 0;
while(i < N-1 && a[i] != x)
  i++;
if(a[i] != x)
  cout << "нет таких";
else
  cout << i + 1;
```

**На алгоритмическом языке**

```
i := 1
нц пока i < N и a[i] <> x
  i := i + 1
кц
если a[i] <> x то
  вывод 'нет таких'
иначе
  вывод i
все
```

**На языке Python**

```
i = 0
while i < n - 1 and a[i] != x:
    i = i + 1
if a[i] != x:
    print("нет таких")
else:
    print(i + 1)
```

**26.**

**Содержание верного ответа и указания к оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

1а. При S=11...30. Во всех этих случаях Петя должен увеличить количество камней в куче в три раза и выиграть. При значениях S<11 невозможно одним ходом (+1, +2 или *3) получить 31 или больше камней.

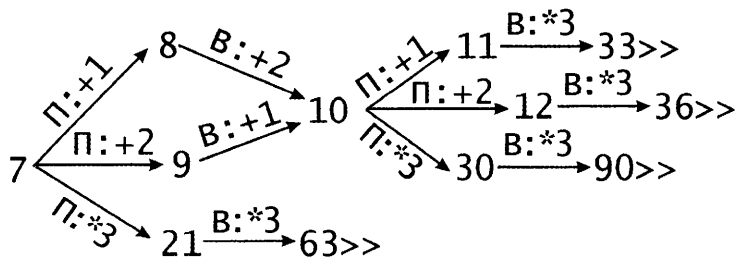
1б. При S=10. Куда бы ни пошел Петя (10+1=11, 10+2=12 или 10*3=30), Ваня утроит число камней в куче и выиграет (11*3=33, 12*3=36 или 30*3=90).

2. S=8 или S=9. В обоих случаях Петя должен получить в кучке 10 камней (8+2=10, 9+1=10). При любом ответном ходе Вани (10+1=11, 10+2=12 или 10*3=30), Петя должен утроить число камней в куче и выиграть (11*3=33, 12*3=36 или 30*3=90).

3. S=7. Если Петя добавит в кучу 1 или 2 камня (7+1=8 или 7+2=9), Ваня должен получить в куче 10 камней (8+2=10, 9+1=10) и в ответ на любой ход Пети (10+1=11, 10+2=12 или 10*3=30), Ваня должен утроить число камней в куче и выиграть (11*3=33, 12*3=36 или 30*3=90). Если Петя утроит число камней в куче (7*3=21), Ваня также должен утроить число камней в куче и выиграть (21*3=63).

Рассмотрим дерево игры (см. рис.).

Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.



**27.**

**Решение задания А**

```
var
    i, j, minp, N : integer;
    a : array [1..10000] of integer; // все элементы
begin
  readln(N);
  for i := 1 to N do
    readln(a[i]);
  minp := 1000*1000+1;
  for i := 1 to n-6 do
    for j := i+6 to n do
      if (a[i] * a[j] mod 2 = 0) and (a[i] * a[j] < minp) then
        minp := a[i] * a[j];
  writeln(minp)
end.
```

**Решение задания Б**

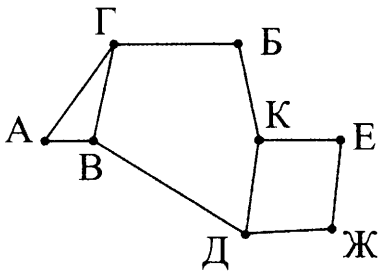
```
var
    i, j, x, minp, min, min2, N : integer;
    a : array [1..6] of integer; // предыдущие введенные 6 элементов
begin
 readln(N);
 for i := 1 to 6 do
   readln(a[i]); // запоминаем первые 5 элементов последовательности
 // начальное значение минимального искомого произведения
 minp := 1000*1000+1;
 min := 1001;  // начальное значение минимального элемента, отстоящего
               // от текущего элемента на расстояние не меньше чем 6
 min2 := 1001; // начальное значение минимального четного элемента,
               // отстоящего от текущего на расстояние не меньше чем 6
 for i := 7 to N do
 begin
   // проверим, не нужно ли сменить min (a[1] лучше?)
   if a[1] < min then
     min := a[1];
   // проверим, не нужно ли сменить min2 (a[1] лучше?)
   if (a[1] < min2) and (a[1] mod 2 = 0) then
     min2 := a[1];
   readln(x); // считываем очередной элемент
   // проверим, не составляет ли текущий элемент лучшую пару, чем имеем
   if x mod 2 = 0 then
   begin
     if x * min < minp then
       minp := x * min
   end
   else
     if x * min2 < minp then
       minp := x * min2;
   for j := 1 to 5 do // сдвигаем массив 6-ти последних элементов на 1
     a[j] := a[j+1];
   a[6] := x          // текущий элемент запоминаем в его "хвосте"
 end;
 writeln(minp)
end.
```

## Вариант 7

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 2 | yzx | 19 | 5363 | 1 | xxD | 17 | 840 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 60 | 62 | 24 | EFCA | 240 | 23 | 18 | 7,16,25 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 2600 | 10 | 7 | 234 | 5 | 28 | 64 | |

Дополнение к ответу задания 3.

| | В | Б | К | Е | Ж | А | Г | Д |
|---|---|---|---|---|---|---|---|---|
| В | | | | | | 10 | 14 | 21 |
| Б | | | 11 | | | | 18 | |
| К | | 11 | | 7 | | | | 23 |
| Е | | | 7 | | 25 | | | |
| Ж | | | | 25 | | | | 12 |
| А | 10 | | | | | | 19 | |
| Г | 14 | 18 | | | | 19 | | |
| Д | 21 | | 23 | | 12 | | | |




**А-Г**

**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | |
|---|---|
| Элементы ответа:<br>1) 6<br>2) 2 –1 2 –3 (или любая последовательность 4-х целых чисел, в которой два числа 2, а остальные 2 числа — неположительные)<br>3) (Исправление программы на языке Паскаль). В строке «`prod := 0;`» должно быть «`prod := 1;`».<br>В строке «`prod := prod + a;`» должно быть «`prod := prod * a;`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!** В задаче требовалось выполнить четыре действия:<br>1. Указать, что выведет программа для указанных входных данных,<br>2. Привести пример входных данных, при которых программа работает верно,<br>3. Исправить одну ошибку в программе,<br>4. Исправить вторую ошибку в программе | |
| Правильно выполнены все пункты задания. Ошибки исправлены верно.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. Ни одна правильная строка не указана в качестве неправильной. | 3 |
| Правильно выполнены три действия из четырёх. Верное указание на ошибку при неверном исправлении при этом не засчитывается. Не более одной верной строки указано как неправильная. | 2 |
| Правильно выполнено два действия из четырёх. Не более одной верной строки указано в качестве неверной. | 1 |
| Правильно выполнено менее двух действий | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
m := 1000;
for i := 1 to N do
  if (a[i] >= 100) and
     (a[i] <= 999) and
     (a[i] mod 10 <> 4) and
     (a[i] < m) then
    m := a[i];
if m <> 1000 then
  writeln(m)
else
  writeln('NO')
```

**На языке Бейсик**

```
M = 1000
FOR I = 1 TO N
IF (A(I) > 99) AND
   (A(I) < 1000) AND
   (A(I) MOD 10 <> 4) AND
   (A(I) < M) THEN
  M = A(I)
END IF
NEXT I
IF M = 1000 THEN
  PRINT "NO"
ELSE
  PRINT M
END IF
```

**На языке C++**

```
m = 1000;
for(i = 0 ; i < N ; i++)
  if( (a[i] > 99) &&
      (a[i] < 1000) &&
      (a[i] % 10 != 4) &&
      (a[i] < m) )
    m = a[i];
if( m < 1000 )
  cout << m;
else
  cout << "NO";
```

**На алгоритмическом языке**

```
m := 1000
нц для i от 1 до N
  если a[i] > 99 и a[i] < 1000 и
       mod(a[i],10)<>4 и a[i] < m
    то
      m := a[i]
  все
кц
если m = 1000 то
  вывод 'NO'
иначе
  вывод m
все
```

**На языке Python**

```
m = 1000
for i in range(0, n):
    if a[i] >= 100 and a[i] <= 999 and a[i] % 10 != 4 and a[i] < m:
        m = a[i]
if m < 1000:
    print(m)
else:
    print("NO")
```

**26.**

**Содержание верного ответа и указания к оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

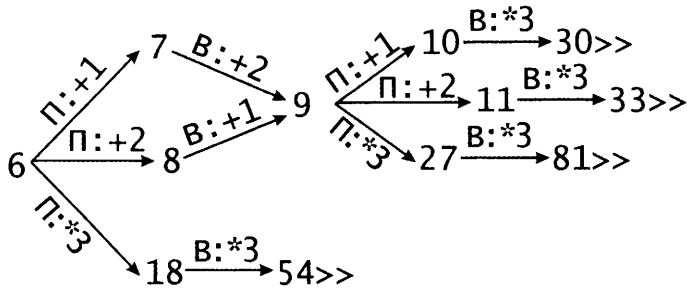
1а. При S=10...29. Во всех этих случаях Петя должен увеличить количество камней в куче в три раза и выиграть. При значениях S<10 невозможно одним ходом (+1, +2 или *3) получить 30 или больше камней.

1б. При S=9. Куда бы ни пошел Петя (9+1=10, 9+2=11 или 9*3=27), Ваня утроит число камней в куче и выиграет (10*3=30, 11*3=33 или 27*3=81).

2. S=7 или S=8 или S=3 (достаточно указать любые два значения из трех. Например, S=7 и S=8). В обоих случаях Петя должен получить в кучке 9 камней (7+2=9, 8+1=9). При любом ответном ходе Вани (9+1=10, 9+2=11 или 9*3=27), Петя должен утроить число камней в куче и выиграть (10*3=30, 11*3=33 или 27*3=81).

3. S=6. Если Петя добавит в кучу 1 или 2 камня (6+1=7 или 6+2=8), Ваня должен получить в куче 9 камней (7+2=9, 8+1=9) и в ответ на любой ход Пети (9+1=10, 9+2=11 или 9*3=27), Ваня должен утроить число камней в куче и выиграть (10*3=30, 11*3=33 или 27*3=81). Если Петя утроит число камней в куче (6*3=18), Ваня также должен утроить число камней в куче и выиграть (18*3=54).

Рассмотрим дерево игры:




Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.

**27.**

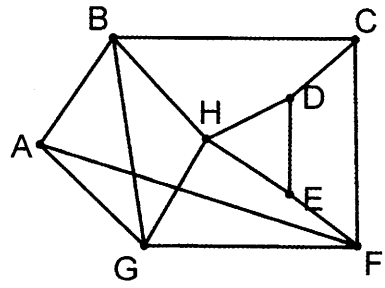
```
var c,h,r:char;
  num:array['A'..'Z']of integer; {массив количества букв алфавита}
  sym:array['A'..'Z']of char;
  k:integer;
begin
 for c:='A' to 'Z' do
 begin
   num[c]:=0;    {обнуляем массив num}
   sym[c]:=c     {заполняем массив sym}
 end;
 read(c);
 while c<>'#' do
 begin
   {если текущий символ - буква}
   if (upcase(c)>='A') and (upcase(c)<='Z') then
     inc(num[upcase(c)]);
   read(c)
 end;
 for h:='Y' downto 'A' do
   for c:='A' to h do
     if num[c]>num[succ(c)] then
     begin
       k:=num[c];              r:=sym[c];
       num[c]:=num[succ(c)];   sym[c]:=sym[succ(c)];
       num[succ(c)]:=k;        sym[succ(c)]:=r
     end;
 h:='A';
 while num[h]=0 do
   inc(h);
 for c:=h to 'Z' do
   write(sym[c])
end.
```

## Вариант 8

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 3 | yzwx | 25 | 3 | 19 | 951 | 2 | 39 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 15 | 20 | 25 | 20 | 300 | 45544 | 30 | 52 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 2 | 99 | 4 | 555 | 113 | 37 | 88 | |

Дополнение к ответу задания 3. (D-G)

| | F | D | A | E | G | C | H | B |
|---|---|---|---|---|---|---|---|---|
| **F** | | | * | * | * | * | | |
| **D** | | | | * | | * | * | |
| **A** | * | | | | * | | | * |
| **E** | * | * | | | | | * | |
| **G** | * | | * | | | | * | * |
| **C** | * | * | | | | | | * |
| **H** | | * | | * | * | | | * |
| **B** | | | * | | * | * | * | |




**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | |
|---|---|
| Элементы ответа:<br>1) 2000<br>2) 1 2 3 4 (или любая последовательность 4-х целых чисел, в которой есть хотя бы одно положительное число)<br>3) (Исправление программы на языке Паскаль)<br>В строке «`if min1 > 0 then`» должно быть «`if min1 < 2000 then`» (или `min1 <> 2000` или `min1 <= 1000`).<br>В строке «`writeln(0)`» должно быть «`writeln('NO')`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!** В задаче требовалось выполнить четыре действия:<br>1. Указать, что выведет программа для указанных входных данных,<br>2. Привести пример входных данных, при которых программа работает верно,<br>3. Исправить одну ошибку в программе,<br>4. Исправить вторую ошибку в программе | |
| Правильно выполнены все пункты задания. Ошибки исправлены верно.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. Ни одна правильная строка не указана в качестве неправильной. | 3 |
| Правильно выполнены три действия из четырёх. Верное указание на ошибку при неверном исправлении при этом не засчитывается. Не более одной верной строки указано как неправильная. | 2 |
| Правильно выполнено два действия из четырёх. Не более одной верной строки указано в качестве неверной. | 1 |
| Правильно выполнено менее двух действий | 0 |
| **Максимальный балл** | **3** |

**25.**

**На языке Паскаль**

```
k := 0;
for i:=1 to N-1 do
  if (a[i] * a[i+1] mod 2 = 0) and
     (a[i] * a[i+1] > 0) then
    k := k + 1;
writeln(k)
```

**На языке Бейсик**

```
K = 0
FOR I = 1 TO N-1
  IF (A(I) * A(I+1) MOD 2 = 0) AND
     (A(I) * A(I+1) > 0) THEN
    K = K + 1
  END IF
NEXT I
PRINT K
```

**На языке C++**

```
k = 0;
for(i = 0 ; i < N-1 ; i++)
  if( a[i] * a[i+1] % 2 == 0 &&
      a[i] * a[i+1] > 0 )
    k++;
cout << k;
```

**На алгоритмическом языке**

```
k := 0
нц для i от 1 до N-1
  если mod(a[i] * a[i+1],2) = 0 и
       a[i] * a[i+1] > 0 то
    k := k + 1
  все
кц
вывод k
```

**На языке Python**

```
k = 0
for i in range(0, n):
    if a[i] * a[i+1] % 2 == 0 and a[i] * a[i+1] > 0:
        k = k + 1
print(k)
```

**26.**

**Содержание верного ответа и указания к оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

**1а)** $21 \leqslant S \leqslant 41$.

**1б)** $S = 11$.

**2)** $S = 20$. Первым ходом Петя должен добавить 1 камень в первую кучу (получится (6,20)). При любом ответном ходе Вани Петя должен удвоить вторую кучу и выиграть.

*Другой вариант верного ответа на задание 2:*

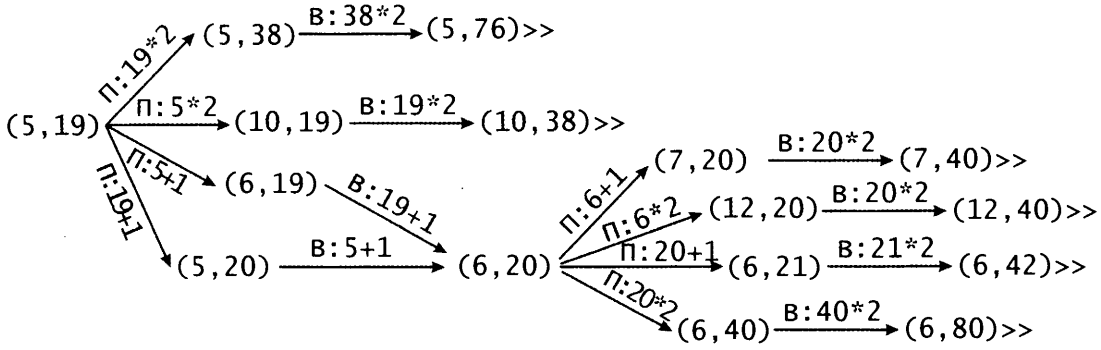
$S = 18$. Первым ходом Петя должен добавить удвоить число камней в первой куче (получится (10,18)). При любом ответном ходе Вани Петя должен удвоить вторую кучу и выиграть.

**3)** $S = 19$.

Если Петя добавит 1 камень в какую-нибудь кучку, Ваня должен добавить 1 камень в другую кучку. Получится ситуация (6,20). При любом ответном ходе Пети Ваня должен удвоить вторую кучу и выиграть.

Если Петя удвоит какую-нибудь кучку, Ваня должен удвоить вторую кучу и выиграть.

Рассмотрим дерево игры:

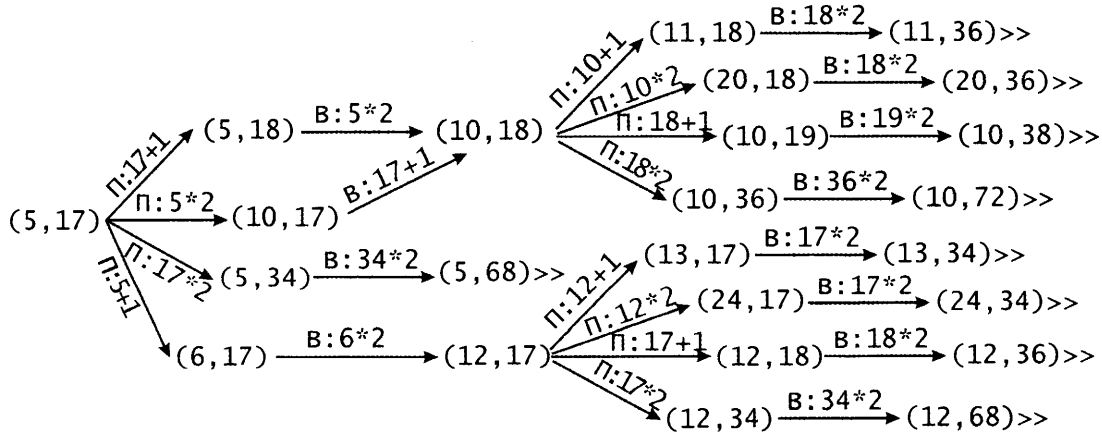



В дереве рассмотрены все возможные ходы Пети и ответные ходы Вани, соответствующие выигрышной стратегии.

Знаком «>>» обозначены позиции, в которых партия заканчивается.

*Другой вариант верного ответа на задание 3:*

$S = 17$.



**27.**

```
var s,k:array[1..99] of integer;
    ch:char;
    i,N,sh,ball,max,nmax:integer;
begin
 for i:=1 to 99 do {обнуляем массивы}
 begin
   s[i]:=0;
   k[i]:=0
 end;
 readln(N); {считали количество строк}
 for i:=1 to N do {перебираем все входные строки}
 begin
   repeat
     read(ch)
   until ch=' ';{считана фамилия}
   repeat
     read(ch)
   until ch=' ';{считано имя}
   readln(sh,ball); {считали номер школы и балл ученика}
   s[sh]:=s[sh]+ball; {считаем сумму баллов по школе}
   k[sh]:=k[sh]+1     {считаем количество учеников из школы}
 end;
 for i:=1 to 99 do
   if k[i]>0 then
     s[i]:=s[i] div k[i]; {считаем средний балл по каждой школе}
 max:=1;
 nmax:=1;
 for i:=2 to 99 do {ищем максимум среди средних баллов}
   if s[i]>s[max] then
   begin
     max:=i;
     nmax:=1
   end
   else
     if s[i]=s[max] then {считаем количество максимумов}
       nmax:=nmax+1;
 if nmax=1 then
   writeln(max,' ',s[max])
 else
   writeln(nmax)
end.
```

## Вариант 9

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 10 | 23 | 12 | 4 | AxxB | 916 | 7 | 35 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| A37 | 192 | 51413121 | 224 | 600 | 877 | 19 | 316 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 66 | 181 | 4 | 631 | 76 | 18 | 68 | |

**24.**

| **Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)** | |
|---|---|
| Элементы ответа:<br>1) 3<br>*Примечание. Программа считает среднее арифметическое чётных элементов. Поэтому результат будет (2+4)/2=3.*<br>2) 2 3 4 0<br>*Правильным ответом является любая последовательность целых чисел, оканчивающаяся нулём, в которой среднее арифметической чётных чисел равно среднему арифметическому нечётных чисел. При этом эти средние арифметические не должны равняться нулю.*<br>3) В программе две ошибки:<br>а) вместо проверки на нечётность числа проверяются на чётность,<br>б) неверно проверяется отсутствие нечётных чисел. Проверка суммы на ноль не верна, например, для последовательности –3 3 0.<br>Исправление программы на языке Паскаль:<br>*Первая ошибка*:<br>Строку «`if a mod 2 = 0 then`» заменить на «`if a mod 2 <> 0 then`».<br>*Вторая ошибка*:<br>В строке «`if s = 0 then`» должно быть «`if c = 0 then`». | |
| **Указания по оцениванию** | **Баллы** |
| **Обратите внимание!** В задаче требовалось выполнить три действия.<br>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.<br>1. Верно указано, что именно выведет программа при указанных в условии входных данных.<br>2. Указаны входные данные, при которых программа работает верно.<br>3. Указана и верно исправлена ошибка, не проверяющая первое введённое число (но проверяющая последнее введённое число).<br>Пункт 3 считается выполненным, если правильно указано, как именно нужно исправить ошибку. | |
| Правильно выполнены все пункты задания. Ошибка исправлена верно. Программа после исправлений для всех последовательностей целых чисел, оканчивающихся нулём, верно определяет среднее арифметическое нечётных элементов последовательности.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения. | 3 |
| Правильно выполнены два действия из трёх. Верное указание на ошибку при неверном исправлении при этом не засчитывается. | 2 |
| Правильно выполнено только одно действие из трёх, то есть, либо только выполнен пункт 1, либо пункт 2, либо верно исправлена ошибка путём её явного указания и исправления или в новом тексте программы | 1 |
| Все пункты задания выполнены неверно или отсутствуют. | 0 |
| **Максимальный балл** | 3 |

**25.**

| **На языке Паскаль** | **На языке Бейсик** |
|---|---|
| `m := 101;`<br>`for i := 1 to n do`<br>`  if (abs(a[i]) <= 100) and`<br>`  (a[i] mod 4 = 0) and`<br>`  (a[i] < m) then`<br>`  m := a[i];`<br>`for i := 1 to n do`<br>`  if not ((abs(a[i]) <= 100) and`<br>`  (a[i] mod 4 = 0)) then`<br>`  a[i] := m;`<br>`for i := 1 to n do`<br>`  write(a[i]);` | `M = 101`<br>`FOR I = 1 TO N`<br>`  IF ABS(A(I)) <= 100 AND`<br>`  A(I) MOD 4 = 0 and`<br>`  A(I) < M THEN`<br>`  M = A(I)`<br>`  END IF`<br>`NEXT I`<br>`FOR I = 1 TO N`<br>`  IF NOT(ABS(A(I)) <= 100 AND`<br>`  A(I) MOD 4 = 0) THEN`<br>`  A(I) = M`<br>`  END IF`<br>`  PRINT A(I)`<br>`NEXT I` |

**На языке C++**

```
m = 101;
for(i = 0 ; i < N ; i++)
  if(abs(a[i]) <= 100 &&
     a[i] % 4 == 0 &&
     a[i] < m)
    m = a[i];
for(i = 0 ; i < N ; i++) {
  if(!(abs(a[i]) <= 100 &&
       a[i] % 4 == 0))
    a[i] = m;
  cout << a[i] << endl;
}
```

**На алгоритмическом языке**

```
m := 101
нц для i от 1 до N
  если abs(a[i]) <= 100 и
       mod(a[i],4)=0 и a[i] < m
    то
      m := a[i]
  все
кц
нц для i от 1 до N
  если не (abs(a[i]) <= 100 и
           mod(a[i],4) = 0)
    то
      a[i] := m
  все
  вывод a[i], нс
кц
```

**На языке Python**

```
m = 101
for i in range(0, n):
    if abs(a[i]) <= 100 and a[i] % 4 == 0 and a[i] < m:
        m = a[i]
for i in range(0, n):
    if not(abs(a[i]) <= 100 and a[i] % 4 == 0):
        a[i] = m
    print(a[i])
```

**26.**

**Содержание верного ответа и указания к оцениванию**

**(допускаются иные формулировки ответа, не искажающие его смысла)**

1. Для обеих позиций (4,30) и (6,29) выигрышную стратегию имеет Ваня. Он выигрывает первым же ходом независимо от хода Пети. Для этого ему необходимо удвоить количество камней во второй куче.

*Обоснование.* Из позиции (4,30) Петя может получить позиции (5,30), (8,30), (4,31) и (4,60). Во всех случаях сумма камней в кучах меньше 65. Однако при ходе Вани (удвоение второй кучи) сумма становится не менее 65.

Из позиции (6,29) Петя может получить позиции (7,29), (12,29), (6,30) и (6,58). Во всех случаях сумма камней в кучах меньше 65. Однако при ходе Вани (удвоение второй кучи) сумма становится не менее 65.

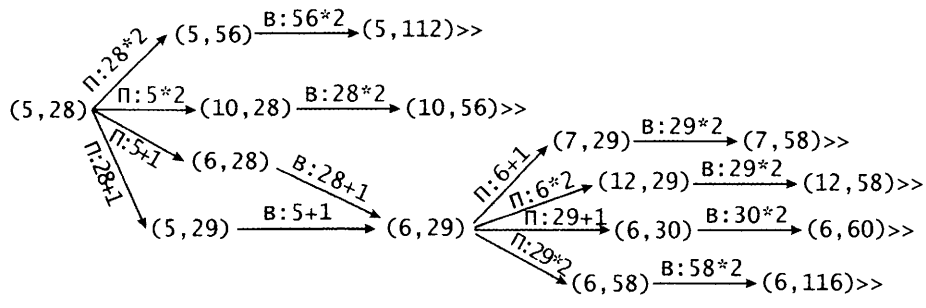
2. Для всех трех позиций (4,29), (6,28) и (5,29) выигрышную стратегию имеет Петя.

Из позиции (4,29) Петя может добавить 1 камень во вторую кучу и получить (4,30).

А из позиций (6,28) и (5,29) Петя может получить позицию (6,29) добавив камень во вторую (6,28+1) или первую (5+1,29) кучу соответственно.

Для полученных позиций (4,30) и (6,29) в пункте 1 доказано, что они проигрышные. Как бы ни походил Ваня, Петя удвоит количество камней во второй куче и выиграет.

3. В позиции (5,28) выигрышную стратегию имеет Ваня. Если Петя первым ходом добавит 1 камень в какую-нибудь кучу, получится позиция (6,28) или (5,29). В пункте 2 доказано, что эти позиции выигрышные. Петя должен добавить 1 камень в другую кучу (6,28+1) или (5+1,29) соответственно и в ответ на любой ответный ход Вани удвоить число камней во второй куче. Если же Петя первым ходом удвоит число камней в какой-нибудь куче (получится (10,28) или (5,56)), Ваня своим первым ходом удвоит число камней во второй куче и выиграет. Рассмотрим дерево игры (см. рис.).




Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.

**27.**

```
var k:array[0..100] of integer; {число учеников, набравших такой балл}
    ch:char;
    i,N,sh,ball,num,s:integer;
begin
 for i:=0 to 100 do {обнуляем массив}
   k[i]:=0;
 readln(N); {считали количество строк}
 for i:=1 to N do {перебираем все входные строки}
 begin
   repeat
     read(ch)
   until ch=' ';{считана фамилия}
   repeat
     read(ch)
   until ch=' ';{считано имя}
   readln(sh,ball); {считали номер школы и балл ученика}
   k[ball]:=k[ball]+1  {считаем количество учеников, набравших такой балл}
 end;
 num:=N div 5; {вычисляем 20% от количества учеников}
 s:=0;
 i:=101;
 while s<num do
 begin
   i:=i-1;
   s:=s+k[i]
 end;
 if s=num then {"отлично" можно поставить ровно 20% участников}
   writeln(i)
 else
   if k[i]=s then {наибольший балл набрало более 20% участников}
     writeln(i)
   else       {ученики, набравшие "i" баллов, не получат "отлично"}
   begin
     i:=i+1;
     while k[i]=0 do {ищем участников с большим баллом}
       i:=i+1;
     writeln(i)
   end
end.
```

## Вариант 10

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 16 | 35 | 12 | 4 | 6 | 78 | 13 | 150 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Б24 | 189 | 7 | 510 | 750 | 5 | 28 | 75 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
| 27 | 24 | 10 | 70 | 16 | 102 | 21 | |

**24.**

**Содержание верного ответа и указания по оцениванию**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

Элементы ответа:

1.

| Область | Условие 1 ($x$>=3) | Условие 2 ($x$<=9) | Условие 3 ($x$>=7) | Программа выведет | Область обрабатывается верно |
|---|---|---|---|---|---|
| A | нет | — | — | — | нет |
| B | да | да | нет | не принадлежит | нет |
| C | да | да | нет | не принадлежит | да |
| D | да | да | да | принадлежит | да |
| E | да | нет | — | — | нет |

2. Возможная доработка (Паскаль):

```
if (x>=3) and (x<=5) or (x>=7) and (x<=9) then
  write('принадлежит')
else
  write('не принадлежит')
```

Возможны и другие способы доработки.

Например:

```
if x>=3 then
  if x<=5 then
    write('принадлежит')
  else
    if x>=7 then
      if x<=9 then
        write('принадлежит')
      else
        write('не принадлежит')
    else
      write('не принадлежит')
else
  write('не принадлежит')
```

| Указания по оцениванию | Баллы |
|---|---|
| **Обратите внимание!** В задаче требуется выполнить **три** действия.<br>1. Заполнить таблицу.<br>2. Исправить ошибку в условном операторе.<br>3. Исправить ошибку, связанную с неправильным набором условий.<br>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.<br>1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.<br>2. Неправильное использование условного оператора, в результате чего при невыполнении первого или второго условия программа не выдавала ничего (отсутствуют случаи ELSE). Исправлением этой ошибки может быть либо добавление случая ELSE к каждому условию IF, либо объединение всех условий IF в одно при помощи конъюнкции.<br>В сложных случаях это действие считается выполненным, если программа выдаёт одно из двух сообщений: «принадлежит» или «не принадлежит» — для любых чисел $x$, при этом программа не стала работать хуже, чем раньше, т.е. для всех точек, для которых программа ранее выдавала верный ответ, доработанная программа также должна выдавать верный ответ.<br>3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие $x \leqslant 5$). Кроме того, необходимо учесть, что области не соединены. Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции, либо использование сложной комбинации каскадных условий.<br>В сложных случаях это действие считается выполненным, если верно определены заштрихованные области, т.е. программа выводит сообщение «принадлежит» для всех точек закрашенных областей, и только для них, для точек вне заштрихованных областей программа выводит «не принадлежит» или не выводит ничего.<br>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения | |

| | |
|---|---|
| Выполнены все три действия | 3 |
| Правильно выполнены два действия из трёх (исправлены обе ошибки, но в пункте 1 задания не приведена таблица (либо таблица содержит ошибки в двух и более строках), либо приведена таблица (которая содержит ошибки не более чем в одной строке), но исправлена только одна ошибка программы).<br>При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо «$x \leqslant 5$» используется «$x<5$» | 2 |
| Правильно выполнено только одно действие из трёх, т.е. либо только приведена таблица, которая содержит ошибки в не более чем одной строке, либо таблица не приведена (или приведена и содержит ошибки более чем в одной строке), но исправлена одна ошибка программы. При оценивании этого задания на 1 балл допускается не учитывать корректность работы программ на точках границ областей (вместо нестрогих неравенств в решении были использованы строгие неравенства или наоборот) | 1 |
| Все пункты задания выполнены неверно (таблица анализа правильности алгоритма не приведена либо содержит ошибки в двух и более строках, программа не приведена либо ни одна из двух ошибок не исправлена) | 0 |
| **Максимальный балл** | 3 |

**25.**

**На языке Паскаль**

```
k := 0;
for i := 1 to N do
  if (a[i] > 9) and
     (a[i] < 100) then
    k := k + 1;
if k > 0 then
  writeln('YES')
else
  writeln('NO')
```

**На языке Бейсик**

```
K = 0
FOR I = 1 TO N
  IF A(I) >= 10 AND
     A(I) <= 99 THEN
    K = K + 1
  END IF
NEXT I
IF K > 0 THEN
  PRINT "YES"
ELSE
  PRINT "NO"
END IF
```

**На языке C++**

```
k = 0;
for(i = 0 ; i < N ; i++)
  if(a[i] > 9 && a[i] < 100)
    k++;
if(k > 0)
  cout << "YES";
else
  cout << "NO";
```

**На алгоритмическом языке**

```
k := 0
нц для i от 1 до N
  если a[i] >= 10 и a[i] <= 99 то
    k := k + 1
  все
кц
если k > 0 то
  вывод 'YES'
иначе
  вывод 'NO'
все
```

**На языке Python**

```
k = 0
for i in range(0, n):
    if a[i] >= 10 and a[i] <= 99:
        k = k + 1
if k > 0:
    print("YES")
else:
    print("NO")
```

**26. Содержание верного ответа**
**(допускаются иные формулировки ответа, не искажающие его смысла)**

Выигрывает второй игрок.

Для доказательства рассмотрим неполное дерево игры, оформленное в виде таблицы, где в каждой ячейке записаны координаты фишки на каждом этапе игры.

| | 1-й ход | 2-й ход | 3-й ход | 4-й ход |
|---|---|---|---|---|
| **Стартовая позиция** | **I игрок (все варианты хода)** | **II игрок (выигрышный ход)** | **I игрок (все варианты хода)** | **II игрок (выигрышный ход, один из вариантов)** |
| 1, 0 | 4, 0 | 7, 3 | 10, 3 | 13, 6 |
| | | | 10, 6 | 13, 6 |
| | | | 7, 6 | 10, 9 |
| | 4, 3 | 7, 3 | 10, 3 | 13, 6 |
| | | | 10, 6 | 13, 6 |
| | | | 7, 6 | 10, 9 |
| | 1, 3 | 4, 6 | 7, 6 | 10, 9 |
| | | | 7, 9 | 10, 9 |
| | | | 4, 9 | 7, 12 |

Таблица содержит все возможные варианты ходов первого игрока. Из неё видно, что при любом ходе первого игрока у второго имеется ход, приводящий к победе.

**27.**

```
var
    i, a, b, s, a1, b1, a2, N : integer;
begin
  a := 0; // кратчайшее расстояние от пункта R до "текущего" съезда
          // на дорогу-связку с дороги A
  b := 0; // кратчайшее расстояние от пункта R до "текущего" съезда
          // на дорогу-связку с дороги B
  readln(N);
  for i := 1 to N do
  begin
    readln(a1,b1,s);
    if a + a1 < b + b1 + s then // если до следующего съезда с дороги A
         // быстрее проехать по дороге A,
         // чем проехать по дороге B и переехать по дороге-связке
      a2 := a + a1              // запомним это расстояние
    else                        // иначе
      a2 := b + b1 + s;         // запомним длину пути по дороге B
    if b + b1 < a + a1 + s then // если до следующего съезда с дороги B
         // быстрее проехать по дороге B,
         // чем проехать по дороге A и переехать по дороге-связке
      b := b + b1               // запомним это расстояние
    else                        // иначе
      b := a + a1 + s;          // запомним длину пути по дороге A
    a := a2 // в переменную a запишем временно хранящийся в a2 результат
  end;
  if a < b then
    writeln(a)
  else
    writeln(b)
end.
```

Единый государственный экзамен

# Бланк ответов № 1

Код региона | Код предмета | Название предмета

С порядком проведения единого государственного экзамена ознакомлен(-а).
Подпись участника ЕГЭ строго внутри окошка

Резерв - 4

**Заполнять гелевой или капиллярной ручкой ЧЕРНЫМИ чернилами ЗАГЛАВНЫМИ ПЕЧАТНЫМИ БУКВАМИ и ЦИФРАМИ по следующим образцам:**

А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z , -
1 2 3 4 5 6 7 8 9 0 À Á Â Ä Ç È É Ê Ë Ì Ï Ù Ú Û Ü ß Ç

**ВНИМАНИЕ!** ***Все бланки и контрольные измерительные материалы рассматриваются в комплекте***

## Результаты выполнения заданий с КРАТКИМ ОТВЕТОМ

| | | | |
|---|---|---|---|
| 1 | | 21 | |
| 2 | | 22 | |
| 3 | | 23 | |
| 4 | | 24 | |
| 5 | | 25 | |
| 6 | | 26 | |
| 7 | | 27 | |
| 8 | | 28 | |
| 9 | | 29 | |
| 10 | | 30 | |
| 11 | | 31 | |
| 12 | | 32 | |
| 13 | | 33 | |
| 14 | | 34 | |
| 15 | | 35 | |
| 16 | | 36 | |
| 17 | | 37 | |
| 18 | | 38 | |
| 19 | | 39 | |
| 20 | | 40 | |

## Замена ошибочных ответов на задания с КРАТКИМ ОТВЕТОМ

| | | | |
|---|---|---|---|
| - | | - | |
| - | | - | |
| - | | - | |

## Заполняется ответственным организатором в аудитории:

Количество заполненных полей «Замена ошибочных ответов»

Подпись ответственного организатора строго внутри окошка

Единый государственный экзамен

**Бланк ответов № 2** | **лист 1**

Код региона | Код предмета | Название предмета

Резерв - 5

Бланк ответов № 2 (лист 2)

Лист

Перепишите значения полей "Код региона", "Код предмета", "Название предмета" из БЛАНКА РЕГИСТРАЦИИ.
Отвечая на задания с РАЗВЕРНУТЫМ ОТВЕТОМ, пишите аккуратно и разборчиво, соблюдая разметку страницы.
Не забудьте указать номер задания, на которое Вы отвечаете, например, 31.
Условия задания переписывать не нужно.

**ВНИМАНИЕ!** ***Все бланки и контрольные измерительные материалы рассматриваются в комплекте***

Оборотная сторона бланка НЕ ЗАПОЛНЯЕТСЯ. Используйте бланк ответов № 2 (лист 2).

*Учебное издание*

Серия «ЕГЭ-2020. Это будет на экзамене»

12+

**Ушаков Денис Михайлович**

# ЕГЭ-2020

# ИНФОРМАТИКА

## 10 тренировочных вариантов экзаменационных работ для подготовки к единому государственному экзамену

***Редакция «Образовательные проекты»***
Ответственный редактор *Н. А. Шармай*
Компьютерная вёрстка *Л. А. Быковой*
Технический редактор *Е. П. Кудиярова*

Подписано в печать 25.06.2019. Формат 60×84 $^{1}/_{8}$. Усл. печ. л. 21,39.
Гарнитуры SchoolBook. Бумага газетная пухлая.
Тираж 3000 экз. Заказ Э-7132.
Отпечатано в типографии ООО «Экопейпер».
420044, Россия, г. Казань, пр. Ямашева, д. 36Б.

Произведено в Российской Федерации
Изготовлено в 2019 г. Изготовитель: ООО «Издательство АСТ»
Общероссийский классификатор продукции ОК-034-2014
(КПЕС 2008); 58.11.1 — книги, брошюры печатные

ООО «Издательство АСТ». 129085, г. Москва, Звёздный бульвар,
дом 21, строение 1, комната 705, пом. I, 7 этаж
Наш электронный адрес: www.ast.ru; e-mail: stelliferovskiy@ast.ru
www.book24.ru

**По вопросам приобретения книг обращаться по адресу:**
123317, г. Москва. Пресненская наб., д. 6, стр. 2, Деловой комплекс «Империя», а/я № 5