

1

СИСТЕМЫ СЧИСЛЕНИЯ. ПЕРЕВОД ЧИСЕЛ МЕЖДУ РАЗЛИЧНЫМИ СИСТЕМАМИ СЧИСЛЕНИЯ

1.1. Сколько единиц в двоичной записи числа 109?

Решение.

Переведём число 109 в двоичную систему счисления.

Способ 1. Будем делить число на 2 с остатком, пока оно не станет равно нулю. Потом посчитаем количество остатков, равных 1.

$$\begin{array}{r}
 109 \div 2 = 54 \text{ (остаток 1)} \\
 54 \div 2 = 27 \text{ (остаток 0)} \\
 27 \div 2 = 13 \text{ (остаток 1)} \\
 13 \div 2 = 6 \text{ (остаток 1)} \\
 6 \div 2 = 3 \text{ (остаток 0)} \\
 3 \div 2 = 1 \text{ (остаток 1)} \\
 1 \div 2 = 0 \text{ (остаток 1)}
 \end{array}$$

$109 = 1101101_2$. Количество единиц = 5.

Ответ: 5.

Способ 2. Разложим число 109 на сумму степеней двойки:

$$\begin{aligned}
 109 &= 64 + 45 = 2^6 + 32 + 13 = 2^6 + 2^5 + 8 + 5 = \\
 &= 2^6 + 2^5 + 2^3 + 4 + 1 = 2^6 + 2^5 + 2^3 + 2^2 + 2^0.
 \end{aligned}$$

Каждому слагаемому в этом разложении соответствует единица в двоичной записи числа. То есть, количество единиц в двоичной записи числа равно количеству слагаемых в разложении на сумму степеней двойки. Имеем 5 слагаемых.

Ответ: 5.

1.1.1. Сколько единиц в двоичной записи числа 197?

Ответ: _____.

1.1.2. Сколько единиц в двоичной записи числа 508?

Ответ: _____.

1.1.3. Сколько единиц в двоичной записи числа 171?

Ответ: _____.

1.1.4. Сколько значащих нулей в двоичной записи числа 188?

Ответ: _____.

1.1.5. Сколько значащих нулей в двоичной записи числа 237?

Ответ: _____.

1.2. Сколько единиц в двоичной записи восьмеричного числа 3216_8 ?

Решение.

Переведём число 3216_8 в двоичную систему счисления. Для этого вместо каждого разряда восьмеричного числа запишем три двоичных разряда. (Три, потому что $8 = 2^3$). Удобнее всего это делать по таблице соответствия между двоичными и восьмеричными числами. Эту таблицу лучше всего знать наизусть.

Если таблица не известна наизусть:

Способ 1. Переведём каждую восьмеричную цифру в двоичную систему счисления. Делением на 2 с остатком или разложением на сумму степеней двойки.

$3 = 2 + 1 = 2^1 + 2^0 = 011_2$. (Не забываем, что вместо каждой восьмеричной цифры нужно записать ровно три двоичные цифры. Поэтому, при меньшем количестве цифр дописываем слева нужное количество нулей).

$2 = 2^1 = 010_2$; $1 = 2^0 = 001_2$; $6 = 4 + 2 = 2^2 + 2^1 = 110_2$.

Получаем всё вместе: $3216_8 = 011\ 010\ 001\ 110_2$. Считаем количество единиц.

Ответ: 6.

Способ 2. Быстро построим таблицу соответствия между двоичными и восьмеричными числами. По таблице запишем вместо каждого восьмеричного числа три двоичных разряда.

Способ 3. (Самый медленный и не рекомендуемый). Переведём исходное число в десятичную систему счисления. После этого результат переведём в двоичную систему счисления.

1.2.1. Сколько единиц в двоичной записи восьмеричного числа 2417_8 ?

Ответ: _____.

1.2.2. Сколько единиц в двоичной записи шестнадцатеричного числа $3B5E_{16}$?

Ответ: _____.

1.2.3. Сколько единиц в двоичной записи восьмеричного числа 263_8 ?

Ответ: _____.

1.2.4. Сколько единиц в двоичной записи шестнадцатеричного числа $5D74_{16}$?

Ответ: _____.

1.2.5. Сколько значащих нулей в двоичной записи шестнадцатеричного числа $6F2C_{16}$?

Ответ: _____.

1.3. Даны 4 целых числа, записанных в шестнадцатеричной системе счисления: CA_{16} , $8E_{16}$, $D3_{16}$, AE_{16} . Сколько среди них чисел, которые меньше чем 315_8 ?

Решение.

Чтобы сравнивать числа, они должны быть в одной системе счисления. Нам даны 4 числа в шестнадцатеричной системе счисления и 1 число в восьмеричной. Проще перевести число из восьмеричной системы в десятичную. Удобнее всего это сделать через двоичную систему счисления.

Переводим 315_8 в двоичную систему счисления. Каждый разряд запишем как 3 двоичных разряда: $011\ 001\ 101_2$. Ведущий ноль можно не писать. Теперь переведём результат в шестнадцатеричную систему счисления. Для этого сгруппируем разряды по 4 справа налево: $1100\ 1101_2$.

Вместо каждой группы их 4-х двоичных разрядов запишем шестнадцатеричную цифру.

$1100_2 = C_{16}$; $1101_2 = D_{16}$. Получаем CD_{16} .

Сравниваем CD_{16} с исходными числами:

$CA_{16} < CD_{16}$? — Да

$8E_{16} < CD_{16}$? — Да

$D3_{16} < CD_{16}$? — Нет

$AE_{16} < CD_{16}$? — Да

Подсчитываем количество верных сравнений.

Ответ: 3.

1.3.1. Даны 4 целых числа, записанные в шестнадцатеричной системе счисления: $B5_{16}$, $9F_{16}$, AC_{16} , $C1_{16}$.

Сколько среди них чисел, которые меньше чем 265_8 ?

Ответ: _____.

1.3.2. Даны 4 целых числа, записанные в восьмеричной системе счисления: 261_8 , 263_8 , 265_8 , 267_8 .

Сколько среди них чисел, которые больше чем $B2_{16}$?

Ответ: _____.

1.3.3. Даны 4 целых числа, записанные в 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: _____.

1.3.4. Даны 4 целых числа, записанные в 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: _____.

1.3.5. Даны 4 целых числа, записанные в различных системах счисления: $A9_{16}$, AE_{16} , $B3_{16}$, AB_{16} .

Сколько среди них чисел, значение которых лежит между 10100110_2 и 265_8 ?

Ответ: _____.

1.4. Сколько существует различных 2-значных чисел в десятичной системе счисления?

Решение.

Способ 1. Рассматриваем всего 2 позиции. На 1-ю позицию можно поставить любую из 9 цифр (всего в десятичной системе счисления 10 цифр, но цифра 0 на первой позиции не может стоять, потому что в этом случае число не будет считаться двузначным). На 2-ю позицию можно поставить любую из 10 цифр (все возможные в десятичной системе счисления). Ответ получается перемножением этих вариантов: 9×10 .

Ответ: 90.

Способ 2. Самое маленькое двузначное число — 10. Самое маленькое трёхзначное число — 100.

Количество двузначных чисел — результат вычитания из числа 100 числа 10. В десятичной системе счисления это будет: $100_{10} - 10_{10} = 90_{10}$. Переводить результат вычитания в десятичную систему счисления не нужно, потому что он уже в этой системе счисления.

Ответ: 90.

1.4.1. Сколько существует различных трёхзначных чисел в десятичной системе счисления?

Ответ: _____.

1.4.2. Сколько существует различных двухзначных чисел в восьмеричной системе счисления?

Ответ: _____.

1.4.3. Сколько существует различных двухзначных чисел в шестнадцатеричной системе счисления?

Ответ: _____.

1.4.4. Сколько существует различных трёхзначных чисел в пятеричной системе счисления?

Ответ: _____.

1.4.5. Сколько существует различных трёхзначных чисел в восьмеричной системе счисления?

Ответ: _____.

1.5. Сколько чисел находится между 375_8 и 403_8 ?

Решение.

Способ 1. Этот способ можно применить, если числа несильно отличаются друг от друга. Выпишем все числа, лежащие между ними:

376_8

377_8

400_8

401_8

402_8

Подсчитаем количество чисел.

Ответ: 5.

❗ При решении задачи этим способом нужно помнить, что в рассматриваемой системе счисления (восьмеричной) бывают только цифры от 0 до 7, т.е. следующим после числа 377_8 будет число 400_8 , а не 378_8 !

Способ 2. Способ работает при любой разнице между числами.

Вычтем («столбиком») из второго числа первое:

$$\begin{array}{r} \cdot \cdot \\ - 403_8 \\ \underline{375_8} \\ 006_8 \end{array}$$

Так как нам нужно количество чисел, находящихся **между** (т.е. не включая границы), отнимем из полученной разности 1.

Ответ: 5.

1.5.1. Сколько чисел находится между 275_8 и 304_8 ?

Ответ: _____.

1.5.2. Сколько чисел находится между $C9_{16}$ и $D4_{16}$?

Ответ: _____.

1.5.3. Сколько чисел находится между 11001001_2 и $E6_{16}$?

Ответ: _____.

1.5.4. Укажите количество различных целых чисел x , которые удовлетворяют условию: $325_8 < x < E7_{16}$.

Ответ: _____.

1.5.5. Укажите количество различных целых чисел x , которые удовлетворяют условию: $634_8 < x < 1F6_{16}$.

Ответ: _____.

1.6. Во сколько раз 76_8 меньше чем 7600_8 ? В ответе запишите только число в десятичной системе счисления.

Решение.

«Во сколько раз одно число больше другого» — это результат деления второго числа на первое. Или, по-другому, «на какое число нужно умножить первое, чтобы получить второе». Чтобы из числа 76_8 получить 7600_8 , нужно дописать к нему справа два нуля, т.е. умножить на 100_8 . Переведём 100_8 в десятичную систему счисления.

Ответ: 64.

1.6.1. Во сколько раз 56_{10} меньше, чем 5600_{10} ? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

1.6.2. Во сколько раз 34_8 меньше, чем 3400_8 ? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

1.6.3. Во сколько раз 101110000_2 больше, чем 10111_2 ? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

1.6.4. Во сколько раз $EB00_{16}$ больше, чем EB_{16} ? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

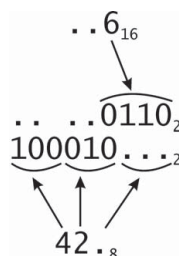
1.6.5. Во сколько раз 1101100000_2 больше, чем $6C_{16}$? В ответе запишите только число в десятичной системе счисления.

Ответ: _____.

1.7. Число X в шестнадцатеричной системе счисления оканчивается на 6_{16} , а в восьмеричной системе счисления — трёхзначное и начинается на 42_8 . Укажите это число в десятичной системе счисления.

Решение.

Переведём имеющиеся у нас числа в двоичную систему счисления.



Точками обозначим цифры, нам не известные. Так как первое и второе двоичное числа — это одно и то же число, получаем, что нам известны все его разряды: 100 010 110₂.

Переведём его в десятичную систему счисления: $2^8 + 2^4 + 2^2 + 2^1 = 256 + 16 + 4 + 2 = 278$.

Ответ: 278.

1.7.1. Число X в шестнадцатеричной системе счисления оканчивается на 9_{16} , а в восьмеричной системе счисления — трёхзначное и начинается на 23_8 . Укажите это число в десятичной системе счисления.

Ответ: _____.

1.7.2. Число X в шестнадцатеричной системе счисления оканчивается на 8_{16} , а в восьмеричной системе счисления — трёхзначное и начинается на 35_8 . Укажите это число в десятичной системе счисления.

Ответ: _____.

1.7.3. Число X в шестнадцатеричной системе счисления оканчивается на 6_{16} , а в восьмеричной системе счисления — трёхзначное и начинается на 32_8 . Укажите это число в десятичной системе счисления.

Ответ: _____.

1.7.4. Число X в шестнадцатеричной системе счисления оканчивается на E_{16} , а в восьмеричной системе счисления — трёхзначное и начинается на 27_8 . Укажите это число в десятичной системе счисления.

Ответ: _____.

1.7.5. Число X в шестнадцатеричной системе счисления оканчивается на 7_{16} , а в восьмеричной системе счисления — трёхзначное и начинается на 26_8 . Укажите это число в десятичной системе счисления.

Ответ: _____.

2

АЛГЕБРА ЛОГИКИ. АНАЛИЗ ТАБЛИЦЫ ИСТИННОСТИ ЛОГИЧЕСКОГО ВЫРАЖЕНИЯ

2.1. Логическая функция F задаётся выражением $x \wedge (\neg z \vee y)$.

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z .

В ответе напишите буквы x, y, z (для некоторых задач — ещё буквы w) в том порядке, в котором идут соответствующие им столбцы: сначала — буква, соответствующая первому столбцу; затем — буква, соответствующая второму столбцу; затем — буква, соответствующая третьему столбцу (для некоторых задач, затем — буква, соответствующая четвёртому столбцу). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Пусть задано выражение $x \rightarrow y$, зависящее от двух переменных x и y , и фрагмент таблицы истинности вида:

		$x \rightarrow y$
0	1	0

Тогда первому столбцу соответствует переменная y , а второму столбцу соответствует переменная x . В ответе нужно написать: yx .

Решение.

Анализируем выражение F . Замечаем, что последняя логическая операция — логическое умножение (\wedge). В таблице истинности даны все варианты, когда выражение F истинно.

Логическое умножение истинно только тогда, когда оба операнда — истина. То есть, истиной должны быть x и $(\neg z \vee y)$.

Переменная x должна быть истиной во всех случаях данной таблицы. То есть, x — это третий столбец. Осталось определить, где y , а где — z . Чтобы отличить эти две переменные друг от друга, рассмотрим строку номер 3 (в первых двух строках эти переменные имеют одинаковые значения). То есть, $(\neg z \vee y)$ должно быть 1 при одной переменной равной 0, а другой — 1. Это возможно только при $z = 0$, а $y = 1$.

Значит, первый столбец — z , второй столбец — y .

Ответ: zyx .

2.1.1. Логическая функция F задаётся выражением

$$z \wedge (y \vee \neg x).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.1.2. Логическая функция F задаётся выражением

$$(\neg x) \wedge ((\neg y) \vee z).$$

На рисунке приведён фрагмент таблицы истинности функции F, содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.1.3. Логическая функция F задаётся выражением

$$y \vee ((\neg z) \wedge x).$$

На рисунке приведён фрагмент таблицы истинности функции F, содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.1.4. Логическая функция F задаётся выражением

$$\neg x \vee ((\neg y) \wedge z).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.1.5. Логическая функция F задаётся выражением

$$(z \rightarrow x) \wedge \neg y.$$

На рисунке приведён фрагмент таблицы истинности функции F, содержащий все наборы аргументов, при которых функция F истинна.

			$(z \rightarrow x) \wedge \neg y$
0	0	0	1
0	0	1	1
1	0	1	1

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.2. Логическая функция F задаётся выражением $(z \vee \neg y) \wedge (z \vee x)$.

На рисунке приведён фрагмент таблицы истинности функции F, содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x , y , z .

Решение.

Эта задача решается примерно тем же способом, что и предыдущие, если в ней сначала преобразовать исходное выражение. А именно: вынести за скобки «общий множитель» по распределительному закону — z . Правильнее будет, вероятно, назвать это «общим слагаемым», потому что в алгебре логики действует закон, который не действует в привычной нам арифметике. А именно: $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$. Этим законом, из правой части в левую часть, мы сейчас и воспользуемся. Здесь $A = z$, $B = \neg y$, $C = x$.

$(z \vee \neg y) \wedge (z \vee x) = z \vee (\neg y \wedge x)$. Заметим теперь, что последняя операция в полученном выражении — логическое сложение (\vee). Она истинна, если хотя бы один операнд — истина. Получаем, что $z = 1$ или $(\neg y \wedge x) = 1$. То есть, при $z = 1$ выражение F должно быть обязательно 1. Так как у нас 3 переменных в выражении, то в таблице истинности всего $2^3 = 8$ строк. В половине из этих 8 строк (т.е. в четырёх) переменная z должна быть равна 1. Ищем столбец, в котором четыре единицы. Это второй столбец. Остаётся понять, какому из двух столбцов (первому и третьему) соответствуют переменные x и y . Для этого рассмотрим ту единственную строку, в которой во втором столбце (z) стоит 0. Это четвёртая строка. Чтобы выражение $(\neg y \wedge x)$ было равно 1 при значениях переменных 0 и 1, необходимо, чтобы было $y = 0$ и $x = 1$. Получаем, что первый столбец — это y , а третий столбец — это x . *Ответ: yzx .*

2.2.1. Логическая функция F задаётся выражением

$$(y \wedge x) \vee (y \wedge \neg z).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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, соответствую-

ющая первому столбцу; затем буква, соответствующая второму столбцу и т.д.).

Буквы в ответе пишете 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: _____.

2.2.2. Логическая функция F задаётся выражением

$$(y \wedge \neg z) \vee (\neg x \wedge \neg z).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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 пишете 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: _____.

2.2.3. Логическая функция F задаётся выражением

$$(z \vee \neg y) \wedge (z \vee x).$$

На рисунке приведён фрагмент таблицы истинности функции F, содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.2.4. Логическая функция F задаётся выражением

$$(\neg x \vee \neg y) \wedge (z \vee \neg y).$$

На рисунке приведён фрагмент таблицы истинности функции F, содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.2.5. Логическая функция F задаётся выражением

$$(x \rightarrow z) \wedge \neg(x \wedge y).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

			$(x \rightarrow z) \wedge \neg(x \wedge y)$
0	0	0	1
0	1	0	1
1	0	0	1
1	0	1	1
1	1	0	1

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.3. Логическая функция F задаётся выражением $y \wedge \neg z \wedge (w \rightarrow x)$. На рисунке приведен фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

				$y \wedge \neg z \wedge (w \rightarrow x)$
0	0	1	1	1
0	0	0	1	1
1	0	1	1	1

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z, w .

Решение.

Логическое выражение $y \wedge \neg z \wedge (w \rightarrow x)$ конъюнктивно. Чтобы оно было истинно, каждый «сомножитель» должен быть истинной. То есть, должно быть $y = 1$, $\neg z = 1$ и $(w \rightarrow x) = 1$. Если $y = 1$, то для y нужно искать столбец, состоящий только из единиц. Это 4-й столбец. Если $\neg z = 1$, то z должен быть равен 0. Столбец из одних нулей — второй. Чтобы для первого и третьего столбцов понять, кто из них соответствует переменной w , а кто — переменной x , найдём строку, где эти столбцы отличаются. Это первая строка. Значит, выражение $(w \rightarrow x)$ должно быть равно 1 при одной переменной равной 0, а другой — 1. Это возможно только при $w = 0$, $x = 1$, т.е. первый столбец — это w , а третий столбец — это x .

Ответ: $wzxy$.

2.3.1. Логическая функция F задаётся выражением

$$x \wedge \neg y \wedge ((\neg z) \rightarrow \neg w).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

				$x \wedge \neg y \wedge ((\neg z) \rightarrow \neg w)$
0	0	1	1	1
0	0	0	1	1
1	0	1	1	1

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z, w .

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: _____.

2.3.2. Логическая функция F задаётся выражением

$$w \wedge \neg x \wedge ((\neg y) \rightarrow \neg z).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

				$w \wedge \neg x \wedge ((\neg y) \rightarrow \neg z)$
0	0	1	0	1
1	0	1	0	1
1	1	1	0	1

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z, w .

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: _____.

2.3.3. Логическая функция F задаётся выражением

$$\neg z \wedge y \wedge (w \rightarrow x).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

				$\neg z \wedge y \wedge (w \rightarrow x)$
0	0	0	1	1
1	0	0	1	1
1	1	0	1	1

Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных x, y, z, w .

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: _____.

2.3.4. Логическая функция F задаётся выражением

$$(x \wedge \neg y) \vee ((\neg x) \wedge z).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

2.3.5. Логическая функция F задаётся выражением

$$(z \wedge x) \vee ((\neg x) \wedge \neg y).$$

На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F истинна.

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

Определите, какому столбцу таблицы истинности функции F 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: _____.

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

				$\neg w \vee (y \wedge x) \vee (\neg z \equiv \neg y)$
0			1	0
	1	0	0	0
0	1	0		0

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

Решение.

Будем решать задачу путём последовательных умозаключений.

1. Данная функция дизъюнктивна и при этом равна нулю в данной таблице. Это возможно только в случае, когда каждое её «слагаемое» равно нулю. То есть, $\neg w = 0$, $(y \wedge x) = 0$ и $(\neg z \equiv \neg y) = 0$.

Чтобы выполнялось $\neg w = 0$, нужно, чтобы $w = 1$. Следовательно, в таблице должен быть столбец, состоящий из одних единиц. Единственный вариант для этого — третий столбец. Значит, в его первой строке стоит тоже 1.

2. Теперь рассмотрим требование $(\neg z \equiv \neg y) = 0$. Это возможно только тогда, когда z и y разные. Среди оставшихся первого, третьего и четвёртого столбцов для переменных z и y не подходит пара «первый и третий», потому что в третьей строке стоят два нуля (а z и y должны быть разными). Также для z и y не подходит пара «третий и четвёртый», потому что во второй строке стоят два нуля. Следовательно, для z и y остаётся только пара «первый и четвёртый». Значит, третий столбец — это переменная x .

3. Так как первый и четвёртый столбцы соответствуют переменным z и y , а они должны быть разными, заполним в них пустые ячейки. В обоих случаях это должна быть единица, потому что соответствующая им цифра другого столбца равна нулю.

Рассмотрим полученную на данный момент таблицу. В ней нижними индексами помечены значения, которые были получены в результате соответствующих умозаключений.

	w_1	x_2		$\neg w \vee (y \wedge x) \vee (\neg z \equiv \neg y)$
0	1 ₁		1	0
1 ₃	1	0	0	0
0	1	0	1 ₃	0

4. Чтобы отличить друг от друга переменные z и y (первый и четвёртый столбцы), нужно сделать вывод из оставшегося нерассмотренным выражения $(y \wedge x) = 0$. Для второй и третьей строки таблицы это выражение выполняется. Необходимо заполнить оставшуюся ячейку. В ней может стоять только цифра 1 (если поставить в эту ячейку 0, то получится две одинаковые строки таблицы истинности (первая и третья), что невозможно по определению таблицы истинности). Получается, чтобы в первой строке выражение $(y \wedge x) = 0$ было верным, должно быть $y = 0$. Среди первого и четвёртого столбцов это верно только для первого столбца. Значит, первый столбец — это y , а четвёртый столбец — это z .

Ответ: $ywxz$.

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

				$(w \wedge z) \vee \neg y \vee (\neg x \equiv \neg w)$
1		0	0	0
	1		0	0
1	0	0		0

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: _____.

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

				$\neg x \wedge (w \equiv \neg z) \wedge (w \rightarrow y)$
	1		1	1
0		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: _____.

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

				$(x \equiv (w \vee z)) \vee (\neg w \vee z) \wedge (\neg z \vee y)$
		1		0
1	1			0
1		1		0

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: _____.

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

				$(\neg w \equiv (y \vee x)) \wedge ((y \wedge \neg x) \vee (x \wedge \neg z))$
		0	0	1
0		0		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: _____.

2.4.5. Вася заполнял таблицу истинности функции $x \vee ((z \rightarrow \rightarrow w) \rightarrow (w \equiv y))$, но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z :

				$x \vee ((z \rightarrow w) \rightarrow (w \equiv y))$
0	1		0	0
	1	0	1	0
1				0

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: _____.

6

**ФОРМАЛЬНОЕ ИСПОЛНЕНИЕ
АЛГОРИТМА, ЗАПИСАННОГО
НА ЕСТЕСТВЕННОМ ЯЗЫКЕ.
СОСТАВЛЕНИЕ ЛИНЕЙНОГО
АЛГОРИТМА ДЛЯ ФОРМАЛЬНОГО
ИСПОЛНИТЕЛЯ**

6.1. 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 167 и может являться результатом работы алгоритма. В ответе это число запишите в десятичной системе.

Решение.

Возьмём минимальное число, которое допустимо по условию («превышает 167»). Это число 168. И переведём его в двоичную систему счисления: $168 = 128 + 40 = 2^7 + 32 + 8 = 2^7 + 2^5 + 2^3 = 10101000_2$.

Исследуем, может ли это число являться результатом работы указанного алгоритма. Так как в результате работы алгоритма к числу N добавляются ещё две двоичные цифры, отбросим от двоичного представления числа 168 две последние цифры. Получим 101010_2 .

Применим к получившемуся числу действия а) и б) алгоритма. Сначала применим действие а). Подсчитаем сумму двоичных цифр числа: $1 + 1 + 1 + 1 = 3$. Остаток от деления суммы на 2 равен 1. Допишем эту 1 к числу: 1010101_2 . Применим к нему действие б). Подсчитаем сумму двоичных цифр числа: $1 + 1 + 1 + 1 + 1 = 4$. Остаток от деления суммы на 2 равен нулю. Допишем этот 0 к числу: 10101010_2 .

Сравним полученный результат с двоичным представлением числа 168 (10101000_2). Так как этот результат оказался больше числа 168, делаем вывод, что, во-первых, число 168 не подходит в качестве ответа. Во-вторых, наименьшее число, которое нас устраивает, это и есть число, полученное в результате совершённых действий а) и б). Переведём это число в десятичный вид: $10101010_2 = 2^7 + 2^5 + 2^3 + 2^1 = 128 + 32 + 8 + 2 = 170$.

Ответ: 170.

Рассмотрим другой случай.

Та же задача, но число R должно превышать 181. Переведём 182 в двоичную систему: 10110110_2 . Отбросим последние 2 разряда. Применим к полученному результату действие а) и действие б). Получим 10110100_2 . Это число меньше 182. Значит, не подходит 182 и не подходит это число. Найдём следующее число, которое подойдёт под условие. Для этого прибавим число 1 к двоичной записи числа 101101_2 (оно получилось при отбрасывании двух последних разрядов от двоичного представления числа 182). После добавления числа 1 получается 101110_2 . Применяем к этому числу действия а) и б). Получаем 10111000_2 . Переводим это число в десятичную систему. Это и есть ответ.

Ответ: 184.

6.1.1. Предлагается некоторая операция над двумя произвольными трёхзначными десятичными числами:

1. Записывается результат сложения значений старших разрядов заданных чисел.

2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе — справа.

3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Определите, какие из предложенных чисел могут быть результатом такой операции. Перечислите в алфавитном порядке буквы, соответствующие этим числам, без пробелов и знаков препинания.

А) 141910

В) 15189

- C) 13127
- D) 1487
- E) 111704

Ответ: _____.

6.1.2. Предлагается некоторая операция над двумя произвольными трёхзначными десятичными числами:

1. Записывается результат сложения значений старших разрядов заданных чисел.

2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе — справа.

3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Определите, какие из предложенных чисел могут быть результатом такой операции.

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

- A) 171412
- B) 121419
- C) 81714
- D) 4809
- E) 16182

Ответ: _____.

6.1.3. Автомат получает на вход два двухзначных 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, В. Результат: В8.

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: _____.

6.1.4. Автомат получает на вход два двухзначных 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: _____.

6.1.5. Автомат получает на вход два двухзначных 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: В, 8. Результат: 8В.

Какие из предложенных чисел могут быть результатом работы автомата?

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

- А) 9В
- В) А5
- С) 2Е
- Д) 112
- Е) DƑ

Ответ: _____.

6.2. 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.

Запишите порядок команд в программе преобразования числа **9** в число **43**, содержащей не более 5 команд, указывая лишь номера команд. (Например, 12122 — это программа

прибавь 1

умножь на 2

прибавь 1

умножь на 2

умножь на 2,

которая преобразует число 1 в 20.)

(Если таких программ более одной, то запишите любую из них.)

Решение. Удобнее всего решать эту задачу «с конца», т.е. пытаться получить число 43. Число 43 нечётное. Поэтому оно не может быть получено действием 2. Значит, последним действием было действие 1 (прибавление 1 к числу 42). Хотя число 42 могло быть получено как из числа 41 прибавлением 1, так и из числа 21 удвоением, предположим, что это было действие удвоения (потому что при этом у нас число изменяется гораздо сильнее, а по условию нам желательно найти самую короткую программу).

Если при решении окажется, что число 21 получить из исходного числа нельзя, мы вернёмся на этот этап и рассмотрим число 41. Будем проводить подобные рассуждения далее. Для удобства оформим это решение как раз «с конца», т.е. справа налево. Запишем справа требуемое число (43) и будем «возвращаться назад», записывая левее от него стрелку и число, из которого оно было получено. Над стрелкой будем записывать операцию, которая получает из левого числа правое.

$$\begin{array}{ccc} & +1 & \\ 42 & \rightarrow & 43 \end{array}$$

Затем сделаем такое же действие с полученным (предыдущим) числом. Для каждого чётного числа будем считать, что оно было получено удвоением. А для каждого нечётного — что увеличением на 1. Так будем делать до тех пор, пока либо не получим исходное число (9), либо не получим число, меньшее исходного. Если получим исходное число, то мы получили требуемый алгоритм. Если же (как в данном случае) получим число, меньшее исходного, значит, мы на каком-то этапе приняли неверное решение:

$$\begin{array}{ccccccc} & \times 2 & & \times 2 & & +1 & & \times 2 & & +1 \\ 5 & \rightarrow & 10 & \rightarrow & 20 & \rightarrow & 21 & \rightarrow & 42 & \rightarrow & 43 \end{array}$$

Найдём последнюю операцию « $\times 2$ » и заменим её на « $+1$ ». В данном случае, этого достаточно для получения ответа. Под каждой «стрелкой» подпишем номер команды, соответствующей выполненному действию (1 — для « $+1$ » и 2 — для « $\times 2$ »). Полученную последовательность номеров запишем в качестве ответа.

$$\begin{array}{ccccccc} & +1 & & \times 2 & & +1 & & \times 2 & & +1 \\ 9 & \rightarrow & 10 & \rightarrow & 20 & \rightarrow & 21 & \rightarrow & 42 & \rightarrow & 43 \\ & 1 & & 2 & & 1 & & 2 & & 1 \end{array}$$

Ответ: 12121.

❗ Так как основными ошибками на ЕГЭ являются ошибки по невнимательности, данный способ оформления решения хорош тем, что позволяет легко проверить полученный результат. В построенной схеме последовательность действий, применяемых к числам, записана в естественном порядке (слева направо). Это позволяет легко и удобно контролировать совершаемые действия и, при случае, быстро обнаружить ошибку.

6.2.1. У исполнителя Удвоитель две команды, которым присвоены номера:

1. умножь на 2,

2. прибавь 1.

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

Запишите порядок команд в программе преобразования числа **4** в число **41**, содержащей не более пяти команд, указывая лишь номера команд.

(Например, 21211 — это программа

прибавь 1

умножь на 2

прибавь 1

умножь на 2

умножь на 2,

которая преобразует число 1 в 20.)

(Если таких программ более одной, то запишите любую из них.)

Ответ: _____.

6.2.2. У исполнителя Утроитель две команды, которым присвоены номера:

1. умножь на 3,

2. прибавь 1.

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

Запишите порядок команд в программе преобразования числа **4** в число **41**, содержащей не более пяти команд, указывая лишь номера команд.

(Например, 21211 — это программа

прибавь 1

умножь на 3

прибавь 1

умножь на 3

умножь на 3,

которая преобразует число 1 в 63.)

(Если таких программ более одной, то запишите любую из них.)

Ответ: _____.

6.2.3. У исполнителя Удвоитель две команды, которым присвоены номера:

1. умножь на 2,

2. прибавь 3.

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

Запишите порядок команд в программе преобразования числа **4** в число **47**, содержащей не более пяти команд, указывая лишь номера команд.

(Например, 21211 — это программа

прибавь 3

умножь на 2

прибавь 3

умножь на 2

умножь на 2,

которая преобразует число 1 в 44.)

(Если таких программ более одной, то запишите любую из них.)

Ответ: _____.

6.2.4. У исполнителя Утроитель две команды, которым присвоены номера:

1. умножь на 3,

2. прибавь 2.

Первая из них утраивает число на экране, вторая — увеличивает его на 2.

Запишите порядок команд в программе преобразования числа **3** в число **65**, содержащей не более пяти команд, указывая лишь номера команд.

(Например, 21211 — это программа

прибавь 2

умножь на 3

прибавь 2

умножь на 3

умножь на 3,

которая преобразует число 1 в 66.)

(Если таких программ более одной, то запишите любую из них.)

Ответ: _____.

6.2.5. У исполнителя Утроитель две команды, которым присвоены номера:

1. умножь на 3,

2. прибавь 4.

Первая из них утраивает число на экране, вторая — увеличивает его на 4.

Запишите порядок команд в программе преобразования числа **1** в число **35**, содержащей не более пяти команд, указывая лишь номера команд.

(Например, 21212 — это программа

прибавь 4

умножь на 3

прибавь 4

умножь на 3

прибавь 4,

которая преобразует число 1 в 61.)

(Если таких программ более одной, то запишите любую из них.)

Ответ: _____.

Общее условие к задачам 6.3.

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.

6.3. Приведём фрагмент кодовой таблицы, используемый в данной задаче:

Буква	Кодовое слово	Примечание
х	000000	сбой
А	100101	
В	100111	
С	000011	
Д	110100	

Исходное сообщение АВС было передано в виде:

1001011 1001110 0000110

И затем было принято в виде:

1001110 0000110 1101000.

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

Решение. Первое, что нужно понять, что примерно половина условия для решения задачи не важна. В частности, не важно, какое сообщение было передано изначально и как это было сделано. Важно только, как происходит обработка принятого сообщения. Для каждого принятого двоичного слова подсчитываем количество единиц. Если оно чётное — отбрасываем последнюю цифру и ищем букву с таким кодовым словом в таблице. Если оно нечётно — значит, произошел «сбой» и декодируем это как букву «х». 1001110 — чётное число единиц. Отбрасываем последнюю цифру: 100111. В таблице это буква В. 0000110 — чётное число единиц. Отбрасываем последнюю цифру: 000011. В таблице это буква С. 1101000 — нечётное число единиц. Это сбой («х»).

Ответ: ВСх.

6.3.1. 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
х	000000	сбой
А	100001	
В	011010	
С	110010	
Д	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: _____.

6.3.2. 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
х	000000	сбой
А	101011	
В	011001	
С	110111	
Д	110110	

Исходное сообщение АВС 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: _____.

6.3.3. 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
х	000000	сбой
А	100011	
В	011101	
С	110100	
Д	110110	

Исходное сообщение ABC было передано в виде:

1000111 0111010 1101001

И затем было принято в виде: 1101001 1010111 1000111

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

Ответ: _____.

6.3.4. 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
х	000000	сбой
А	000011	
В	101101	
С	100100	
Д	010110	

Исходное сообщение ABC было передано в виде:

0000110 1011010 1001001

И затем было принято в виде: 1011010 0101101 1011011

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

Ответ: _____.

6.3.5. 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
х	000000	сбой
А	111010	
В	010111	
С	100110	
Д	011100	

Исходное сообщение ABC было передано в виде:
1110100 0101110 1001101

И затем было принято в виде: 1100111 1001101 0111001

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

Ответ: _____.

6.4. 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 выдает результат 156.

Решение.

Число 156 может быть получено записью друг за другом чисел 1 и 56, либо чисел 15 и 6. Первый вариант не подходит, потому что в нем нарушается условие «в порядке невозрастания». Рассмотрим второй вариант. По условию оба числа (15 и 6) получаются сложением двух пар цифр, при этом вторая цифра исходного числа участвует в обеих суммах. При этом, чтобы искомое число было наименьшим, нужно поставить в качестве его первой цифры самое маленькое возможное число.

Попробуем поставить цифру 1.

Тогда в качестве второй цифры нужно будет поставить либо 15 ($16 - 1$), либо 5 ($6 - 1$). Цифры «15» не бывает.

Если второй цифрой сделать 5, тогда третья цифра должна быть 10 ($15 - 5$). Которой тоже не бывает.

Если первой цифрой поставить 2, получится ещё хуже: второй цифрой будет либо 13 ($15 - 2$), либо 4 ($6 - 2$). Цифры «13» не бывает.

Если второй цифрой сделать 4, то третьей цифрой должна быть цифра 11 ($15 - 4$), которой тоже не бывает.

Заметим, что варианты первой цифры 3, 4 и 5 приводят к тому, что в качестве последней цифры получаются всё возрастающие числа 12, 13 и 14.

Если на первое место поставить цифру 6, на втором месте нужно будет поставить цифру 0 ($6 - 6$) или 9 ($15 - 6$).

Если второй поставить 0, то третьей цифрой должна быть цифра 15 ($15 - 0$). Такого не бывает.

Если второй цифрой поставить цифру 9, то третьей цифрой должна быть цифра «-3» ($6 - 9$). Такого также не бывает.

В этот момент может показаться, что рассуждения зашли в тупик. Однако, в действительности, осталось перебрать всего 3 варианта. Варианты первой цифры 7 и 8 требуют второй цифрой 8 и 7, и третья цифрой будет отрицательная.

Поставим на первое место 9.

На втором месте должна будет быть 6. Тогда на третьей позиции получится 0.

Значит, несмотря на весьма большое трёхзначное число, единственный вариант ответа в этой задаче: 960.

Ответ: 960.

Возможно, представление в виде таблицы всех этих размышлений окажется более понятным.

Вариант первой цифры	Вариант второй цифры	Вариант третьей цифры	Примечание
1	14 (15 – 1)	—	Цифры «14» не бывает
	5 (6 – 1)	10 (15 – 5)	Цифры «10» не бывает
2	13 (15 – 2)	—	Цифры «13» не бывает
	4 (6 – 2)	11 (15 – 4)	Цифры «11» не бывает
3	12 (15 – 3)	—	Цифры «12» не бывает
	3 (6 – 3)	12 (15 – 3)	Цифры «12» не бывает
4	11 (15 – 4)	—	Цифры «11» не бывает
	2 (6 – 4)	13 (15 – 2)	Цифры «13» не бывает
5	10 (15 – 5)	—	Цифры «10» не бывает
	1 (6 – 5)	14 (15 – 1)	Цифры «14» не бывает
6	9 (15 – 6)	–3 (6 – 9)	Цифры «–3» не бывает
	0 (6 – 6)	15 (15 – 0)	Цифры «15» не бывает
7	8 (15 – 7)	–2 (6 – 8)	Цифры «–2» не бывает
	–1 (6 – 7)	—	Цифры «–1» не бывает
8	7 (15 – 8)	–1 (6 – 7)	Цифры «–1» не бывает
	–2 (6 – 8)	—	Цифры «–2» не бывает
9	6 (15 – 9)	0 (6 – 6)	Это ответ
	–3 (6 – 9)	—	Цифры «–3» не бывает

Ещё один способ решения. Пары цифр числа образуют суммы: 15 и 6. Так как цифры лежат в диапазоне от 0 до 9, рассмотрим возможные результаты получения числа 15. Их всего четыре: $6 + 9$, $7 + 8$, $8 + 7$ и $9 + 6$. Одну из этих цифр (6, 7, 8 и 9) нужно прибавить к чему-то, чтобы получить вторую сумму — 6. Единственный вариант — цифра 6 ($6 + 0$) и разложение числа 15 как $9 + 6$. Получаем, что мы ищем число, у которого средняя цифра — 6 (она участвует в обеих суммах — как $9 + 6$ и как $6 + 0$), а по краям стоят цифры 9 и 0. То есть, вариант 069 и вариант 960. Вариант 069 не подходит, потому что это не трёхзначное число.

Ответ: 960.

6.4.1. 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: _____.

6.4.2. 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: _____.

6.4.3. 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: _____.

6.4.4. 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: _____.

6.4.5. 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 наименьшее число, при обработке которого автомат выдаёт результат 312.

Ответ: _____.

7

ЭЛЕКТРОННЫЕ ТАБЛИЦЫ. ФОРМУЛЫ. АБСОЛЮТНАЯ И ОТНОСИТЕЛЬНАЯ АДРЕСАЦИЯ ПРИ КОПИРОВАНИИ ФОРМУЛ. ДИАГРАММЫ

7.1. В электронной таблице значение формулы =СРЗНАЧ (В2 : С4) равно 4. Чему равно значение формулы =СУММ (В1 : С4), если значение формулы =СРЗНАЧ (В1 : С1) равно 3?

Решение. Представим себе, как в таблице расположены ячейки:

	А	В	С
1			
2			
3			
4			

В формулы СРЗНАЧ (В2 : С4) = 4 определим сумму ячеек диапазона В2 : С4. Всего в этом диапазоне 6 ячеек. Их среднее арифметическое равно 4. Значит, их сумма равна $6 \cdot 4 = 24$. Так же из формулы СРЗНАЧ (В1 : С1) = 3 определим сумму ячеек диапазона В1 : С1: $2 \cdot 3 = 6$. Теперь можно найти сумму ячеек диапазона В1 : С4. Это $24 + 6 = 30$.

Ответ: 30.

7.1.1. В электронной таблице значение формулы =СРЗНАЧ (А3 : D3) равно 5. Чему равно значение формулы =СУММ (А3 : С3), если значение ячейки D3 равно 6?

Ответ: _____.

7.1.2. В электронной таблице значение формулы =СУММ (А7 : С7) равно 9. Чему равно значение формулы =СРЗНАЧ (А7 : D7), если значение ячейки D7 равно 7?

Ответ: _____.

7.1.3. В электронной таблице значение формулы =СРЗНАЧ (А3 : D4) равно 5. Чему равно значение формулы =СРЗНАЧ (А3 : С4), если значение формулы =СУММ (D3 : D4) равно 4?

Ответ: _____.

7.1.4. В электронной таблице значение формулы

=СРЗНАЧ (С2 : D5) равно 3. Чему равно значение формулы
=СУММ (С5 : D5) , если значение формулы =СРЗНАЧ (С2 : D4) равно 5?

Ответ: _____.

7.1.5. В электронной таблице значение формулы

=СРЗНАЧ (А4 : D5) равно 5. Чему равно значение формулы
=СРЗНАЧ (А4 : С5) , если значение формулы =СУММ (D4 : D5) равно 16?

Ответ: _____.

7.2. При работе с электронной таблицей в ячейку D4 записана формула: = $\$E1+C\2 . Какой вид приобретет формула, после того как ячейку D4 скопируют в ячейку C6?

Решение.

Для начала поясним имеющуюся ситуацию. При копировании формулы из ячейки в ячейку возникает эффект «автоиндексации». То есть, индексы адреса ячейки изменяются на величину, равную величине «смещения» формулы относительно ее начального положения. Индексы — это элементы адреса (буква — номер столбца и число — номер строки). Например, если формулу =C5, находящуюся в ячейке B3, скопировать в ячейку D2, то она станет формулой =E4. Это получилось потому, что смещение из B3 в D2 по номерам столбцов — «+2», а по номерам строк — «-1». Изменяем на эти величины индексы C5. Сдвиг на 2 буквы к букве C даёт букву E, а вычитание 1 из числа 5 даёт 4.

Если же эффект автоиндексации нежелателен, перед тем индексом, который при копировании не должен меняться, ставят знак «\$».

Теперь рассмотрим решаемую задачу. Из ячейки D4 копируют в C6. При этом по столбцам происходит смещение влево на 1 (то есть, «-1»), а по строкам — смещение вниз на 2 (то есть, «+2»). В исходной формуле индексы \$E и \$2 не меняются. Получаем: = $\$E3+A\2 .

Ответ: = $\$E3+A\2 .

7.2.1. При работе с электронной таблицей в ячейку В4 записана формула: $=\$C3*2$.

Какой вид приобретёт формула после того, как ячейку В4 скопируют в ячейку D2?

Ответ: _____.

7.2.2. При работе с электронной таблицей в ячейку С5 записана формула: $=E3+1$.

Какой вид приобретёт формула после того, как ячейку С5 скопируют в ячейку В6?

Ответ: _____.

7.2.3. При работе с электронной таблицей в ячейку С3 записана формула: $=\$B3+C\2 .

Какой вид приобретёт формула после того, как ячейку С3 скопируют в ячейку D2?

Ответ: _____.

7.2.4. При работе с электронной таблицей в ячейку Е5 записана формула: $=\$C3-B\3 .

Какой вид приобретёт формула после того, как ячейку Е5 скопируют в ячейку D3?

Ответ: _____.

7.2.5. При работе с электронной таблицей в ячейку D4 записана формула: $=3-C\$5$.

Какой вид приобретёт формула после того, как ячейку D4 скопируют в ячейку В6?

Ответ: _____.

7.3. Дан фрагмент электронной таблицы.

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

Чему станет равным значение ячейки D1, если в неё скопировать формулу из ячейки C2?

Примечание: знак \$ обозначает абсолютную адресацию.

Решение.

При копировании из ячейки C2 в ячейку D1 номер столбца меняется на «+1», а номер строки меняется на «-1». В формуле =B\$1+\$B3 индексы \$1 и \$B не изменятся. Индекс B изменится на «+1» и станет C. Индекс 3 изменится на «-1» и станет 2. Получим формулу =C\$1+\$B2. Вычислим её. В ячейке C1 находится формула =A2+B1. A2=5, B1=2. Значит, C1=5+2=7. B2=6. Вычислим D1=C1+B2=7+6=13.

Ответ: 13.

7.3.1. Дан фрагмент электронной таблицы.

	A	B	C	D
1	4	5	=A2+B\$1	
2	2	1	=A1+B2	
3	7	6	3	

Чему станет равным значение ячейки D2, если в неё скопировать формулу из ячейки C1?

Примечание: знак \$ обозначает абсолютную адресацию.

Ответ: _____.

7.3.2. Дан фрагмент электронной таблицы.

	A	B	C	D
1	4	5	3	
2	2	1	=A1+B2	
3	6	7	=A\$2+B3	

Чему станет равным значение ячейки D2, если в неё скопировать формулу из ячейки C3?

Примечание: знак \$ обозначает абсолютную адресацию.

Ответ: _____.

7.3.3. 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: _____.

7.3.4. 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 неё скопировать формулу из ячейки B2?

Примечание: знак \$ обозначает абсолютную адресацию.

Ответ: _____.

7.3.5. Дан фрагмент электронной таблицы.

	A	B	C	D
1	7	5	3	4
2	4	=C2+\$C2	1	2
3		=C1+D1	6	7

Чему станет равным значение ячейки A3, если в неё скопировать формулу из ячейки B2?

Примечание: знак \$ обозначает абсолютную адресацию.

Ответ: _____.

7.4. Дан фрагмент электронной таблицы:

	A	B	C	D
1	6		2	
2	=C1+1	=C1*3	=A1/2	=A1+B1+C1

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



Решение.

Вычислим значения тех ячеек, все данные для которых известны:

$$A2 = C1 + 1 = 2 + 1 = 3$$

$$B2 = C1 \cdot 3 = 2 \cdot 3 = 6$$

$$C2 = A1 / 2 = 6 / 2 = 3$$

Анализируем диаграмму и замечаем, что в ней два одинаковых сектора, еще один в два раза больше и ещё один в 4 раза больше. Среди вычисленных значений ячеек имеем два одинаковых ($A2 = C2 = 3$) и ещё одну в 2 раза больше. Значит, значение оставшейся ячейки (D2) должно быть в 4 раза больше одинаковых. То есть, $3 \cdot 4 = 12$. Получаем уравнение:

$$D2 = 12 \Rightarrow A1 + B1 + C1 = 12 \Rightarrow 6 + B1 + 2 = 12 \Rightarrow B1 = 4.$$

Ответ: 4.

7.4.1. 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: _____.

7.4.2. 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?

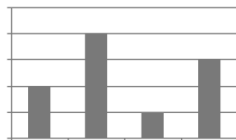


Ответ: _____.

7.4.3. Дан фрагмент электронной таблицы:

	A	B	C	D
1	3		5	
2	$= C1-3$	$= (A1+C1) / 2$	$= A1 / 3$	$= (B1+A2) / 2$

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



Ответ: _____.

7.4.4. Дан фрагмент электронной таблицы:

	A	B	C	D
1	2		6	
2	$= C1 / 2$	$= (A1+1) / 3$	$= (B1+A2) / 4$	$= C1-3$

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



Ответ: _____.

7.4.5. 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: _____.

7.5. Дан фрагмент электронной таблицы:

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

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



Решение.

Обозначим требуемое значение B1 за x . Выразим все значения ячеек диапазона A2:D2.

A2	B2	C2	D2
$(x-3)/2$	$x-3$	$x-6$	$x+3$

Все полученные значения зависят от x (значения неизвестной ячейки B1). Напрямую их не вычислить.

Анализируем диаграмму. Видим из неё, что из четырёх ячеек две одинаковые, третья вдвое больше первых двух, а четвертая вчетверо их больше. Соотнесём с этим значения, полученные при вычислении ячеек A2:D2. Замечаем, что $D2 > B2 > C2$. Значит, D2 имеет самое большое значение, B2 вдвое меньше, а C2 ещё вдвое меньше. Любое из этих соотношений позволяет составить уравнение для нахождения x . Возьмём условие $B2 = 2 \cdot C2$.

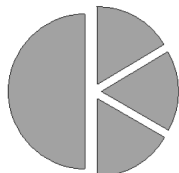
Подставим в него значения: $x-3 = 2 \cdot (x-6) \Rightarrow x-3 = 2x-12 \Rightarrow \Rightarrow 12-3 = 2x-x \Rightarrow 9 = x$.

Ответ: 9.

7.5.1. 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?

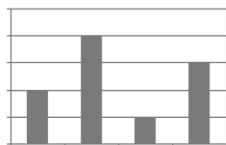


Ответ: _____.

7.5.2. Дан фрагмент электронной таблицы:

	A	B	C	D
1	4			
2	=B1/2	=A2+A1	=A2/2	=A2+C2

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

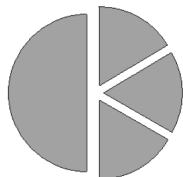


Ответ: _____.

7.5.3. Дан фрагмент электронной таблицы:

	A	B	C	D
1	6			
2	=C2/3	=B1/4	=A1+B2	=A1-B2

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



Ответ: _____.

7.5.4. Дан фрагмент электронной таблицы:

	A	B	C	D
1				9
2	=D1-B2	=B1/3	=B1-B2	=C2/4

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



Ответ: _____.

7.5.5. Дан фрагмент электронной таблицы:

	A	B	C	D
1			8	
2	=B1/2	=C1-A2	=D2-A2-B2	=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

ИСПОЛНЕНИЕ ЦИКЛИЧЕСКОГО АЛГОРИТМА, ЗАПИСАННОГО НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ

8.1. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 0 s = 5 WHILE s < 165 s = s + 40 k = k + 3 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 0; s := 5; while s < 165 do begin s := s + 40; k := k + 3 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 0; s = 5; while (s < 165) { s = s + 40; k = k + 3; } cout << k << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 0 s := 5 нц пока s < 165 s := s + 40 k := k + 3 кц вывод k кон </pre>
Python	
<pre> k = 0 s = 5 while s < 165: s = s + 40 k = k + 3 print(k) </pre>	

Решение.

Способ 1. Подробная трассировка программы.

Трассировка программы — это пошаговое выполнение программы с отслеживанием значений всех переменных. Трассировка бывает подробной и упрощенной.

Рассмотрим подробную сортировку данной программы на языке Паскаль.

Оператор	Условие	Переменные		Пояснения
		k	s	
k:=0;		0		Переменная k получает начальное значение, равное нулю.
s:=5;			5	Переменная s получает начальное значение, равное 5.
while s<165 do	5<165? Да			Начинается цикл с предусловием while. Так как условие выполняется, выполняем тело цикла.
s:=s+40;			45	Значение переменной s (5) увеличивается на 40.
k:=k+3		3		Значение переменной k (0) увеличивается на 3.
while s<165 do	45<165? Да			Снова проверяется условие цикла while. Так как условие выполняется, выполняем тело цикла.
s:=s+40;			85	Значение переменной s (45) увеличивается на 40.
k:=k+3		6		Значение переменной k (3) увеличивается на 3.
while s<165 do	85<165? Да			Снова проверяется условие цикла while. Так как условие выполняется, выполняем тело цикла.
s:=s+40;			125	Значение переменной s (85) увеличивается на 40.
k:=k+3		9		Значение переменной k (6) увеличивается на 3.
while s<165 do	125<165? Да			Снова проверяется условие цикла while. Так как условие выполняется, выполняем тело цикла.

Оператор	Условие	Переменные		Пояснения
		k	s	
<code>s:=s+40;</code>			165	Значение переменной <i>s</i> (125) увеличивается на 40.
<code>k:=k+3</code>		12		Значение переменной <i>k</i> (9) увеличивается на 3.
<code>while s<165 do</code>	165<165? Нет			Снова проверяется условие цикла <code>while</code> . Так как условие не выполняется, цикл заканчивается.
<code>write(k)</code>				На экране — текущее значение переменной <i>k</i> (12).

Ответ: 12.

Способ 2. Упрощенная трассировка программы.

В отличие от подробной трассировки при упрощенной трассировке записываются только текущие значения переменных. Все остальные действия: отслеживание текущей выполняемой строки программы, и результаты проверки условий — выполняются в уме. Этим методом рекомендуется пользоваться для убыстрения выполнения трассировки. Однако это имеет смысл только в том случае, если вы хорошо освоили подробную трассировку и уверены в том, что правильно совершите те действия, которые при упрощенной трассировке выполняются в уме.

Для выполнения трассировки рисуют вертикальную таблицу, в которой столько же столбцов, сколько переменных используется в программе.

Переменные		Пояснения
k	s	
0		После выполнения оператора « <code>k := 0;</code> » переменная <i>k</i> получает начальное значение, равное нулю.
	5	После выполнения оператора « <code>s := 5;</code> » переменная <i>s</i> получает начальное значение, равное 5.
	45	Начинается цикл с предусловием <code>while</code> . Так как условие цикла выполняется ($5 < 165$), выполняем тело цикла. После выполнения оператора « <code>s := s + 40;</code> » значение переменной <i>s</i> (5) увеличивается на 40.
3		После выполнения оператора « <code>k := k + 3;</code> » значение переменной <i>k</i> (0) увеличивается на 3.

Переменные		Пояснения
k	s	
	85	Снова проверяется условие цикла <code>while</code> . Так как условие цикла ($45 < 165$) выполняется, выполняем тело цикла. Значение переменной <code>s</code> (45) увеличивается на 40.
6		Значение переменной <code>k</code> (3) увеличивается на 3.
	125	Снова проверяется условие цикла ($85 < 165$). Так как условие выполняется, выполняем тело цикла. Значение переменной <code>s</code> (85) увеличивается на 40.
9		Значение переменной <code>k</code> (6) увеличивается на 3.
	165	Значение переменной <code>s</code> (125) увеличивается на 40.
12		Значение переменной <code>k</code> (9) увеличивается на 3.
		Снова проверяется условие цикла ($165 < 165$). Так как условие не выполняется, цикл заканчивается. На экране — текущее значение переменной <code>k</code> (12).

Способ 3.

Заметим, что в приведенном примере задачи цикл заканчивается достаточно быстро. В реальных заданиях количество выполнений тела цикла может быть существенно больше. Если вы хорошо понимаете, как работает программа, имеет смысл воспользоваться еще более быстрым способом.

Анализируем, что делает программа. Замечаем, что цикл выполняется в зависимости от значения переменной `s`. Начиная с начального значения, равного 5, на каждом шаге к переменной `s` прибавляется одно и то же число (40) до тех пор, пока выполняется условие $s < 165$. Попробуем определить, сколько раз при этом будет выполняться цикл. Найдем значение переменной `s`, при котором условие цикла перестанет выполняться. Прибавляя к начальному значению 5 число 40, переменная `s` станет равна 165. Определим, сколько раз нужно прибавлять к числу 5 число 40, чтобы получить 165: $(165 - 5) / 40 = 4$.

❗ Если последние рассуждения не очевидны, можно решить задачу алгебраически. Изменение переменной `s` начиная с 5 на каждом шаге на 40 — это арифметическая прогрессия. Начальное значение — 5, разность — 40.

Используем формулу: $s_n = s_1 + d(n - 1)$. $s_n = 5 + 40(n - 1)$.

Найдём, при каком n перестанет выполняться условие $s_n < 165$. То есть, станет $s_n \geq 165$: $5 + 40(n - 1) \geq 165 \Rightarrow 40(n - 1) \geq 160 \Rightarrow n - 1 \geq 160 / 40 \Rightarrow n - 1 \geq 4 \Rightarrow n \geq 5$. Значит, условие перестанет выполняться при $n = 5$. То есть, было сделано 4 увеличения переменной s на 40.

Мы вычислили, что цикл выполнится 4 раза. На каждом шаге цикла к переменной k добавляется число 3. То есть, за 4 шага цикла переменная k увеличится на 12. Начальное значение переменной k было равно 0. Значит, после выполнения цикла переменная k станет равна $12 (0 + 12)$.

Ответ: 12.

8.1.1. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 0 s = 5 WHILE s < 205 s = s + 10 k = k + 1 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 0; s := 5; while s < 205 do begin s := s + 10; k := k + 1 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 0; s = 5; while (s < 205) { s = s + 10; k = k + 1; } cout << k << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 0 s := 5 нц пока s < 205 s := s + 10 k := k + 1 кц вывод k кон </pre>

Python
<pre>k = 0; s = 5 while s < 205: s = s + 10; k = k + 1 print(k)</pre>

Ответ: _____.

8.1.2. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER k = 0 s = 6 WHILE s < 605 s = s + 30 k = k + 3 WEND PRINT k</pre>	<pre>var k, s : integer; begin k := 0; s := 6; while s < 605 do begin s := s + 30; k := k + 3 end; write(k) end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int k, s; k = 0; s = 6; while (s < 605) { s = s + 30; k = k + 3; } cout << k << endl; return 0; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> k, s k := 0 s := 6 <u>нц пока</u> s < 605 s := s + 30 k := k + 3 <u>кц</u> <u>вывод</u> k <u>кон</u></pre>
Python	
<pre>k = 0; s = 6 while s < 605: s = s + 30; k = k + 3 print(k)</pre>	

Ответ: _____.

8.1.3. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 2 s = 8 WHILE s < 500 s = s + 20 k = k + 5 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 2; s := 8; while s < 500 do begin s := s + 20; k := k + 5 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s k := 2 s := 8 нц пока s < 500 s := s + 20 k := k + 5 кц вывод k кон </pre>
Python	
<pre> k = 2 s = 8 while s < 500: s = s + 20 k = k + 5 print(k) </pre>	

Ответ: _____.

8.1.4. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 1 s = 3 WHILE s < 100 s = s + 5 k = k + 2 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 1; s := 3; while s < 100 do begin s := s + 5; k := k + 2 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s k := 1 s := 3 нц пока s < 100 s := s + 5 k := k + 2 кц вывод k кон </pre>
Python	
<pre> k = 1 s = 3 while s < 100: s = s + 5 k = k + 2 print(k) </pre>	

Ответ: _____.

8.1.5. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 3 s = 5 WHILE s < 120 s = s + 6 k = k + 3 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 3; s := 5; while s < 120 do begin s := s + 6; k := k + 3 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 3; s = 5; while (s < 120) { s = s + 6; k = k + 3; } cout << k << endl; return 0; } </pre>	<pre> нач цел k, s k := 3 s := 5 <u>нц пока</u> s < 120 s := s + 6 k := k + 3 <u>кц</u> <u>вывод</u> k <u>кон</u> </pre>
Python	
<pre> k = 3 s = 5 while s < 120: s = s + 6 k = k + 3 print(k) </pre>	

Ответ: _____.

8.2. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 2 s = 10 WHILE s < 168 s = s + 20 k = k * 2 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 2; s := 10; while s < 168 do begin s := s + 20; k := k * 2; end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 2; s = 10; while (s < 168) { s = s + 20; k = k * 2; } cout << k << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 2 s := 10 нц пока s < 168 s := s + 20 k := k * 2 кц вывод k кон </pre>
Python	
<pre> k = 2 s = 10 while s < 168: s = s + 20 k = k * 2 print(k) </pre>	

Решение.

Подобно задаче 8-1, эту задачу можно решать трассировкой или аналитически. Воспользуемся аналитическим способом. В программе выполняется цикл с условием. Он выполняется до тех пор, пока переменная *s*, начиная с начального значения 10, и увеличиваясь на каждом шаге на 20, не станет больше или равна 168.

При этом переменная k , начиная с начального значения, равного 2, удваивается на каждом шаге. Сначала найдём, сколько раз выполнится цикл. По формуле арифметической прогрессии:
 $10 + 20(n - 1) \geq 168 \Rightarrow 20(n - 1) \geq 158 \Rightarrow n - 1 \geq 158 / 20 \Rightarrow n \geq 7,95 + 1 \Rightarrow n \geq 8,95$. Это случится при $n = 9$. Значит, цикл выполнится 8 шагов. Значит, $k = 2 \cdot 2^8 = 512$.

Ответ: 512.

8.2.1. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER k = 1 s = 8 WHILE s < 208 s = s + 20 k = k * 2 WEND PRINT k</pre>	<pre>var k, s : integer; begin k := 1; s := 8; while s < 208 do begin s := s + 20; k := k * 2 end; write(k) end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int k, s; k = 1; s = 8; while (s < 208) { s = s + 20; k = k * 2; } cout << k << endl; return 0; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> k, s k := 1 s := 8 <u>нц пока</u> s < 208 s := s + 20 k := k * 2 <u>кц</u> <u>вывод</u> k <u>кон</u></pre>
Python	
<pre>k = 1; s = 8 while s < 208: s = s + 20; k = k * 2 print(k)</pre>	

Ответ: _____.

8.2.2. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 2 s = 7 WHILE s < 70 s = s + 10 k = k * 2 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 2; s := 7; while s < 70 do begin s := s + 10; k := k * 2 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 2; s = 7; while (s < 70) { s = s + 10; k = k * 2; } cout << k << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 2 s := 7 нц пока s < 70 s := s + 10 k := k * 2 кц вывод k кон </pre>
Python	
<pre> k = 2 s = 7 while s < 70: s = s + 10 k = k * 2 print(k) </pre>	

Ответ: _____.

8.2.3. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 0 s = 1 WHILE s < 1000 s = s * 2 k = k + 5 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 0; s := 1; while s < 1000 do begin s := s * 2; k := k + 5 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 0; s = 1; while (s < 1000) { s = s * 2; k = k + 5; } cout << k << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 0 s := 1 нц пока s < 1000 s := s * 2 k := k + 5 кц вывод k кон </pre>
Python	
<pre> k = 0 s = 1 while s < 1000: s = s * 2 k = k + 5 print(k) </pre>	

Ответ: _____.

8.2.4. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 1000 s = 2 WHILE s < 600 s = s * 2 k = k - 5 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 1000; s := 2; while s < 600 do begin s := s * 2; k := k - 5 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 1000; s = 2; while (s < 600) { s = s * 2; k = k - 5; } cout << k << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 1000 s := 2 нц пока s < 600 s := s * 2 k := k - 5 кц вывод k кон </pre>
Python	
<pre> k = 1000 s = 2 while s < 600: s = s * 2 k = k - 5 print(k) </pre>	

Ответ: _____.

8.2.5. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 50 s = 2 WHILE s < 1000 s = s * 2 k = k - 3 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 50; s := 2; while s < 1000 do begin s := s * 2; k := k - 3 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 50; s = 2; while (s < 1000) { s = s * 2; k = k - 3; } cout << k << endl; return 0; } </pre>	<pre> нач цел k, s k := 50 s := 2 нц пока s < 1000 s := s * 2 k := k - 3 кц вывод k кон </pre>
Python	
<pre> k = 50 s = 2 while s < 1000: s = s * 2 k = k - 3 print(k) </pre>	

Ответ: _____.

8.3. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 0 FOR k = 5 TO 8 s = s + k NEXT k PRINT s</pre>	<pre>var k, s : integer; begin s := 0; for k := 5 to 8 do s := s + k; write(s) end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int k, s; s = 0; for(k = 5 ; k <= 8 ; k++) s = s + k; cout << s << endl; return 0; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> k, s s := 0 <u>нц для</u> k <u>от</u> 5 <u>до</u> 8 s := s + k <u>кц</u> <u>вывод</u> s <u>кон</u></pre>
Python	
<pre>s = 0 for k in range(5,9): s = s + k print(s)</pre>	

Решение.

Подобно задаче 8-1, эту задачу можно решать трассировкой или аналитически. Воспользуемся аналитическим методом. Изучая программу, определяем, что в ней в цикле от 5 до 8 к переменной *s* прибавляется значение счетчика цикла. То есть, в переменной *s* оказывается сумма чисел от 5 до 8. Воспользуемся формулой суммы арифметической прогрессии (сумма равна «первый плюс последний пополам, умноженное на количество элементов»). Посчитаем количество элементов от 5 до 8: $8 - 5 + 1 = 4$. Значит, сумма будет равна: $(5 + 8) \cdot 4 / 2 = 13 \cdot 2 = 26$.

Ответ: 26.

8.3.1. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER s = 0 FOR k = 3 TO 8 s = s + k NEXT k PRINT s </pre>	<pre> var k, s : integer; begin s := 0; for k := 3 to 8 do s := s + k; write(s) end. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s s := 0 нц для k от 3 до 8 s := s + k кц вывод s кон </pre>
Python	
<pre> s = 0 for k in range(3,9): s = s + k print(s) </pre>	

Ответ: _____.

8.3.2. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER s = 5 FOR k = 2 TO 9 s = s + k NEXT k PRINT s </pre>	<pre> var k, s : integer; begin s := 5; for k := 2 to 9 do s := s + k; write(s) end. </pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int k, s; s = 5; for(k = 2 ; k <= 9 ; k++) s = s + k; cout << s << endl; return 0; }</pre>	<pre><u>нач</u> <u>цел</u> k, s s := 5 <u>нц</u> <u>для</u> k <u>от</u> 2 <u>до</u> 9 s := s + k <u>кц</u> <u>вывод</u> s <u>кон</u></pre>
Python	
<pre>s = 5 for k in range(2,10): s = s + k print(s)</pre>	

Ответ: _____.

8.3.3. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 1 FOR k = 4 TO 7 s = s * k NEXT k PRINT s</pre>	<pre>var k, s : integer; begin s := 1; for k := 4 to 7 do s := s * k; write(s) end.</pre>
C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> k, s s := 1 <u>нц</u> <u>для</u> k <u>от</u> 4 <u>до</u> 7 s := s * k <u>кц</u> <u>вывод</u> s <u>кон</u></pre>

Python
<pre>s = 1 for k in range(4,8): s = s * k print(s)</pre>

Ответ: _____.

8.3.4. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 2 FOR k = 3 TO 6 s = s * k NEXT k PRINT s</pre>	<pre>var k, s : integer; begin s := 2; for k := 3 to 6 do s := s * k; write(s) end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int k, s; s = 2; for(k = 3 ; k <= 6 ; k++) s = s * k; cout << s << endl; return 0; }</pre>	<pre><u>нач</u> <u>цел</u> k, s s := 2 <u>нц для</u> k <u>от</u> 3 <u>до</u> 6 s := s * k <u>кц</u> <u>вывод</u> s <u>кон</u></pre>
Python	
<pre>s = 2 for k in range(3,7): s = s * k print(s)</pre>	

Ответ: _____.

8.3.5. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER s = 3 FOR k = 5 TO 9 s = s + 2 * k NEXT k PRINT s</pre>	<pre>var k, s : integer; begin s := 3; for k := 5 to 9 do s := s + 2 * k; write(s) end.</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int k, s; s = 3; for(k = 5 ; k <= 9 ; k++) s = s + 2 * k; cout << s << endl; return 0; }</pre>	<pre><u>нач</u> <u>цел</u> k, s s := 3 <u>нц для</u> k <u>от</u> 5 <u>до</u> 9 s := s + 2 * k <u>кц</u> <u>вывод</u> s <u>кон</u></pre>
Python	
<pre>s = 3 for k in range(5,10): s = s + 2 * k print(s)</pre>	

Ответ: _____.

Задачи **8.4** немного отличаются по условию от задач 8.1, но решаются подобно им. Автор уверен, что Вы сможете разобраться в небольших особенностях этих задач самостоятельно.

8.4.1. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 50 s = 1024 WHILE s > 0 s = s \ 2 k = k - 3 WEND PRINT k </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s k := 50 s := 1024 нц пока s > 0 s := div(s, 2) k := k - 3 кц вывод k кон </pre>
Python	
<pre> k = 50 s = 1024 while s > 0: s = s // 2 k = k - 3 print(k) </pre>	

Ответ: _____.

8.4.2. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 3 s = 512 WHILE s > 1 s = s \ 2 k = k + 4 WEND PRINT k </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> <u>алг</u> <u>нач</u> <u>цел</u> k, s k := 3 s := 512 <u>нц</u> <u>пока</u> s > 1 s := div(s,2) k := k + 4 <u>кц</u> <u>вывод</u> k <u>кон</u> </pre>
Python	
<pre> k = 3 s = 512 while s > 1: s = s // 2 k = k + 4 print(k) </pre>	

Ответ: _____.

8.4.3. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 5 s = 512 WHILE s > 5 s = s \ 2 k = k + 4 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 5; s := 512; while s > 5 do begin s := s div 2; k := k + 4 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 5; s = 512; while (s > 5) { s = s / 2; k = k + 4; } cout << k << endl; return 0; } </pre>	<pre> нач цел k, s k := 5 s := 512 нц пока s > 5 s := div(s, 2) k := k + 4 кц вывод k кон </pre>
Python	
<pre> k = 5 s = 512 while s > 5: s = s // 2 k = k + 4 print(k) </pre>	

Ответ: _____.

8.4.4. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 25 s = 256 WHILE s > 6 s = s \ 2 k = k - 2 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 25; s := 256; while s > 6 do begin s := s div 2; k := k - 2 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 25; s = 256; while (s > 6) { s = s / 2; k = k - 2; } cout << k << endl; return 0; } </pre>	<pre> нач цел k, s k := 25 s := 256 нц пока s > 6 s := div(s, 2) k := k - 2 кц вывод k кон </pre>
Python	
<pre> k = 25 s = 256 while s > 6: s = s // 2 k = k - 2 print(k) </pre>	

Ответ: _____.

8.4.5. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 1024 s = 50 WHILE s > 30 s = s - 4 k = k \ 2 WEND PRINT k </pre>	<pre> var k, s : integer; begin k := 1024; s := 50; while s > 30 do begin s := s - 4; k := k div 2 end; write(k) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 1024; s = 50; while (s > 30) { s = s - 4; k = k / 2; } cout << k << endl; return 0; } </pre>	<pre> нач цел k, s k := 1024 s := 50 нц пока s > 30 s := s - 4 k := div(k, 2) кц вывод k кон </pre>
Python	
<pre> k = 1024 s = 50 while s > 30: s = s - 4 k = k // 2 print(k) </pre>	

Ответ: _____.

8.5. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 30 s = 0 WHILE s + k > 15 s = s + 8 k = k - 12 WEND PRINT s </pre>	<pre> var k, s : integer; begin k := 30; s := 0; while s + k > 15 do begin s := s + 8; k := k - 12 end; write(s) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 30; s = 0; while (s + k > 15) { s = s + 8; k = k - 12; } cout << s << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 30 s := 0 нц пока s + k > 15 s := s + 8 k := k - 12 кц вывод s кон </pre>
Python	
<pre> k = 30 s = 0 while s + k > 15: s = s + 8 k = k - 12 print(s) </pre>	

Решение. Как и ранее, эту задачу можно решать трассировкой и аналитически. В данном случае количество шагов цикла невелико, и выполнение трассировки будет не слишком долгим, но при этом даст более надёжный ответ. Однако выполнение трассировки мы уже подробно разобрали в задаче 8.1. Поэтому мы рассмотрим здесь аналитическое решение этой задачи. Для начала определим, сколько раз будет выполняться цикл.

Условием продолжения цикла является условие $s + k > 15$. Изначально $k = 30, s = 0$, Поэтому условие выполняется ($30 + 0 > 15$). Замечаем, что на каждом шаге цикла переменная s увеличивается на 8, а переменная k уменьшается на 12. То есть, сумма $s + k$ на каждом шаге цикла уменьшается на 4. При начальном значении суммы 30 через 4 шага цикла сумма перестанет быть больше 15 (для определения этого количества можно рассмотреть величину $(30 - 15) / 4$). Итак, цикл выполнится 4 раза, и к начальному значению переменной s (0) на каждом шаге добавляется 8. Значит, после цикла значение переменной s будет равно $0 + 4 \cdot 8 = 32$.

Ответ: 32.

8.5.1. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 50 s = 0 WHILE s + k > 10 s = s + 5 k = k - 15 WEND PRINT s </pre>	<pre> 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. </pre>
С++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s k := 50 s := 0 нц пока s + k > 10 s := s + 5 k := k - 15 кц вывод s кон </pre>

Python
<pre>k = 50; s = 0 while s + k > 10: s = s + 5; k = k - 15 print(s)</pre>

Ответ: _____.

8.5.2. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre>DIM k, s AS INTEGER k = 90 s = 0 WHILE s + k > 20 s = s + 5 k = k - 15 WEND PRINT s</pre>	<pre>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.</pre>
C++	Алгоритмический язык
<pre>#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; }</pre>	<pre>алг нач цел k, s k := 90 s := 0 нц пока s + k > 20 s := s + 5 k := k - 15 кц вывод s кон</pre>
Python	
<pre>k = 90; s = 0 while s + k > 20: s = s + 5; k = k - 15 print(s)</pre>	

Ответ: _____.

8.5.3. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 70 s = 0 WHILE s + k < 120 s = s + 15 k = k - 5 WEND PRINT s </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s k := 70 s := 0 нц пока s + k < 120 s := s + 15 k := k - 5 кц вывод s кон </pre>
Python	
<pre> k = 70 s = 0 while s + k < 120: s = s + 15 k = k - 5 print(s) </pre>	

Ответ: _____.

8.5.4. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 100 s = 0 WHILE s + k < 180 s = s + 25 k = k - 10 WEND PRINT s </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> алг нач цел k, s k := 100 s := 0 нц пока s + k < 180 s := s + 25 k := k - 10 кц вывод s кон </pre>
Python	
<pre> k = 100 s = 0 while s + k < 180: s = s + 25 k = k - 10 print(s) </pre>	

Ответ: _____.

8.5.5. Определите, что будет напечатано в результате выполнения следующего фрагмента программы:

Бейсик	Паскаль
<pre> DIM k, s AS INTEGER k = 80 s = 0 WHILE s + k < 160 s = s + 15 k = k - 10 WEND PRINT s </pre>	<pre> var k, s : integer; begin k := 80; s := 0; while s + k < 160 do begin s := s + 15; k := k - 10 end; write(s) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int k, s; k = 80; s = 0; while (s + k < 160) { s = s + 15; k = k - 10; } cout << s << endl; return 0; } </pre>	<pre> алг нач цел k, s k := 80 s := 0 нц пока s + k < 160 s := s + 15 k := k - 10 кц вывод s кон </pre>
Python	
<pre> k = 80 s = 0 while s + k < 160: s = s + 15 k = k - 10 print(s) </pre>	

Ответ: _____.

9

СКОРОСТЬ ПЕРЕДАЧИ ИНФОРМАЦИИ. КОДИРОВАНИЕ ЗВУКОВОЙ И ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

9.1. Имеется неупакованное растровое изображение размером 4096×512 пикселей. Какое максимальное количество цветов может быть в палитре изображения, чтобы в неупакованном виде оно занимало на диске не более 200 Кбайт?

Решение.

Воспользуемся формулой $V = H \cdot W \cdot i / 8$, где V — размер файла с изображением в байтах, H и W — высота и ширина изображения в пикселях, i — количество бит в одном пикселе. Подставим известные величины в формулу: $200 \text{ Кбайт} \geq 4096 \cdot 512 \cdot i / 8 \Rightarrow 200 \cdot 2^{13} \text{ бит} \geq 2^{12} \cdot 2^9 \cdot i / 2^3 \Rightarrow i \leq 200 \cdot 2^{13} / 2^{18} \Rightarrow i \leq 200 / 32 \Rightarrow i \leq 25 / 4$. То есть, наибольшее целое $i = 6$. Наибольшее количество цветов в палитре изображения найдём по формуле: $2^{\text{число_бит}} = \text{число_цветов}$. Подставляем: $2^6 = 64$.

Ответ: 64.

9.1.1. Скольких различных цветов (наибольшее количество) могут быть пиксели неупакованного растрового изображения, имеющего размер 1024×256 пикселей и занимающего на диске 160 килобайт?

Ответ: _____.

9.1.2. Какой объём на диске (в Мбайт) будет занимать неупакованное 16-цветное изображение, содержащее изображение размером 2048×1024 пикселей? В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.1.3. Какой минимальный объём памяти (в Кбайт) нужно зарезервировать, чтобы можно было сохранить любое растровое изображение размером 64×256 пикселей при условии, что

в изображении могут использоваться 4 различных цвета? В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.1.4. Какое максимальное количество цветов может быть в палитре неупакованного растрового изображения, имеющего размер 1024×256 пикселей и занимающего на диске не более 165 Кбайт?

Ответ: _____.

9.1.5. 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: _____.

9.2. Производится двухканальная (стерео) звукозапись с частотой дискретизации 32 кГц и 8-битным разрешением. Запись длится четыре минуты, 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.

Решение.

Воспользуемся формулой для вычисления объёма звукового файла: $V = Ch \cdot v \cdot t \cdot i / 8$, где V — размер файла с изображением в байтах, Ch — количество каналов записи звука, v — частота дискретизации (Гц), t — время звукозаписи (секунд), i — разрешение записи (количество бит в одном измерении уровня). По условию: $Ch = 2$ (стерео), $v = 32\,000$ (32 кГц), $t = 4 \cdot 60 = 240$ (4 минуты), $i = 8$ бит. Подставим эти величины в формулу: $V = 2 \cdot 32\,000 \cdot 240 \cdot 8 / 8 = 15360000$. Переведём эту величину в Мбайты. Для этого поделим её на 1024 дважды. Получится чуть больше 15.

Ответ: 15.

9.2.1. 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: _____.

9.2.2. Производится четырёхканальная (квадро) звукозапись с частотой дискретизации 4 кГц и 32-битным разрешением. Запись длится 1 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: _____.

9.2.3. Производится одноканальная (моно) звукозапись с частотой дискретизации 8 кГц и 16-битным разрешением. Запись длится 2 минуты, 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: _____.

9.2.4. Производится одноканальная (моно) звукозапись с частотой дискретизации 16 кГц и количеством уровней квантования 256. Запись длится 3 минуты, 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: _____.

9.2.5. 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: _____.

9.3. Производится звукозапись музыкального фрагмента в формате стерео (двухканальная запись) с частотой дискретизации 32 кГц и 16-битным разрешением. Результаты записываются в файл, сжатие данных не производится; размер полученного файла 80 Мбайт. Затем производится повторная запись этого же фрагмента в формате квадрато (четырёхканальная запись) с частотой дискретизации 48 кГц и 24-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.

Решение.

Воспользуемся формулой для вычисления объёма звукового файла: $V = Ch \cdot v \cdot t \cdot i / 8$, где V — размер файла с изображением в байтах, Ch — количество каналов записи звука, v — частота дискретизации (Гц), t — время звукозаписи (секунд), i — разрешение записи (количество бит в одном измерении уровня). В этой формуле все величины, от которых зависит объём звукового файла, ему прямо пропорциональны. То есть, например, если какая-нибудь величина увеличится в 2 раза, объём файла также увеличится в два раза. Будем решать задачу исходя из этой пропорциональной зависимости.

Исходный файл был 80 Мбайт. Из двухканальной записи сделали четырёхканальную. Значит, размер файла изменился в $4/2$ раз. Частота дискретизации была 32 кГц, стала 48 кГц. Значит, размер файла изменился в $48/32$ раз. Разрешение звукозаписи было 16 бит, стало 24 бита. Значит, размер файла изменился в $24/16$ раз. Подсчитываем все вместе: $80 \text{ Мбайт} \cdot 4/2 \cdot 48/32 \cdot 24/16 = 80 \text{ Мбайт} \cdot 2 \cdot 3/2 \cdot 3/2 = 360 \text{ Мбайт}$.

Ответ: 360.

9.3.1. Производится звукозапись музыкального фрагмента в формате квадрато (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 производится. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.3.2. Производится звукозапись музыкального фрагмента в формате квадрo (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 производится. Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.3.3. Музыкальный фрагмент был записан в формате 7.1 (восьмиканальная запись), оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла — 48 Мбайт. Затем тот же музыкальный фрагмент был оцифрован повторно в формате стерео (двухканальная запись) с разрешением в 1,5 раза меньше и частотой дискретизации в 2 раза больше, чем в первый раз. 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: _____.

9.4.4. Музыкальный фрагмент был записан в формате стерео (двухканальная запись), оцифрован и сохранён в виде файла без использования сжатия данных. Размер полученного файла — 16 Мбайт. Затем тот же музыкальный фрагмент был оцифрован повторно в формате моно с разрешением в 1,5 раза выше и частотой дискретизации в 2 раза меньше, чем в первый раз. 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: _____.

9.3.5. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город А по каналу связи за 40 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 2 раза меньше и частотой дискретизации в 1,5 раза

больше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город Б. Пропускная способность канала связи с городом Б в 3 раза выше, чем с городом А. Сколько секунд длилась передача файла в город Б?

В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.4. У Васи есть высокоскоростной доступ к сети Интернет со скоростью 2^{24} бит/сек.

Петин компьютер связан с Васиным через канал связи со скоростью 2^{18} бит/сек. Петя договорился с Васей, что тот будет скачивать для него данные объёмом 32 Мбайт из Интернета и ретранслировать их Пете. Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 4 Мбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей? В ответе запишите только целое число, единицу измерения писать не нужно.

Решение.

Вся передача данных по времени разбивается на два этапа. На первом этапе компьютер Васи должен получить первые 4 Мбайт. На втором этапе компьютер Васи начинает передавать данные компьютеру Пети, при этом продолжая получать из Интернета оставшуюся часть данных. Так как канал связи между компьютерами Васи и Пети медленнее, чем канал связи компьютера Васи с Интернетом, то время второго этапа будет зависеть только от времени передачи данных между компьютерами Васи и Пети (Васин компьютер в каждый момент времени уже будет иметь ту информацию, которую нужно передавать Пете). То есть, во время второго этапа по каналу Вася-Петя нужно будет передать все требуемые 32 Мбайт. Посчитаем отдельно время каждого этапа. Воспользуемся формулой: $v = l / t$. Здесь: v — скорость передачи информации (бит в секунду), l — объём передаваемой информации (бит), t — скорость передачи информации (секунд). Из этой формулы: $t = l / v$. Для первого этапа: $t_1 = 4 \cdot 2^{23} / 2^{24} = 2$ секунды. Для второго этапа: $t_2 = 32 \cdot 2^{23} / 2^{18} = 2^5 \cdot 2^{23} / 2^{18} = 2^8 / 2^{18} = 2^{10} = 1024$ секунды. Складываем время обоих этапов.

Ответ: 1026.

9.4.1. У Васи есть высокоскоростной доступ к сети Интернет со скоростью 2^{22} бит/сек.

Петин компьютер связан с Васиным через канал связи со скоростью 2^{19} бит/сек. Петя договорился с Васей, что тот будет скачивать для него данные объёмом 8 Мбайт из Интернета и ретранслировать их Пете. Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 512 Кбайт этих данных. Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей? В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.4.2. Вася имеет высокоскоростной доступ в сеть Интернет при помощи спутниковой тарелки. Пропускная способность канала получения данных со спутника составляет 8 Мбит/сек. Но эта связь работает только в одну сторону (на приём). Для того, чтобы компьютер Васи мог отдавать команды о том, какую информацию нужно передавать со спутника, Вася подключает к компьютеру сотовый телефон, который может передавать в сеть информацию со скоростью не более 256 Кбит/сек. Вася хочет скачать файл объёмом 40 Мбайт. Информация со спутника поступает на компьютер Васи фрагментами не более 1 Мбайта. Для получения каждого фрагмента компьютер Васи должен сначала передать в сеть набор инструкций суммарным объёмом 64 Кбайт.

За какое минимально возможное число секунд Вася может получить весь файл? В ответе запишите только целое число, единицу измерения писать не нужно.

Ответ: _____.

9.4.3. У Васи есть высокоскоростной доступ в Интернет со скоростью 2^{17} бит/сек. Петин компьютер связан с Васиным через канал связи со скоростью 2^{15} бит/сек. Петя договорился с Васей, что тот будет скачивать для него данные объёмом 5 Мбайт из Интернета и ретранслировать их Пете. Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 1024 Кбайт этих данных.

Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей?

Ответ: _____.

9.4.4. У Васи есть высокоскоростной доступ в Интернет со скоростью 2^{19} бит/сек. Петин компьютер связан с Васиным через канал связи со скоростью 2^{14} бит/сек. Петя договорился с Васей, что тот будет скачивать для него данные объёмом 20 Мбайт из Интернета и ретранслировать их Пете. Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 2048 Кбайт этих данных.

Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей?

Ответ: _____.

9.4.5. У Васи есть высокоскоростной доступ в Интернет со скоростью 2^{18} бит/сек. Петин компьютер связан с Васиным через канал связи со скоростью 2^{14} бит/сек. Петя договорился с Васей, что тот будет скачивать для него данные объёмом 5 Мбайт из Интернета и ретранслировать их Пете. Компьютер Васи может начать ретрансляцию данных не раньше, чем им будут получены первые 1024 Кбайт этих данных.

Каков минимально возможный промежуток времени (в секундах) с момента начала скачивания Васей данных до полного их получения Петей?

Ответ: _____.

9.5. Документ объёмом 8 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 50% исходного;
- время, требуемое на сжатие документа, — 15 секунд, на распаковку — 7 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.

Решение.

Сначала вычислим время передачи документа способом Б (без использования архиватора). По формуле $t = l / v: 8 \cdot 2^{23} / 2^{21} = 2^{26} / 2^{21} = 2^5 = 32$ секунды. Так как объём сжатого документа составляет 50% от исходного, а скорость передачи при способе А такая же, что и при способе Б, то и время передачи способом А будет оставлять 50% от времени передачи способом Б. То есть, $32 \cdot 50\% = 16$ секунд. Добавим к этому время на сжатие и время на распаковку: $16 + 15 + 7 = 38$ секунд. Получаем, что способ Б быстрее на $38 - 32 = 6$ секунд.

Ответ: Б6.

9.5.1. 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% от 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.

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

Ответ: _____.

9.5.2. 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% от 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.

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

Ответ: _____.

9.5.3. 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% от 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.

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

Ответ: _____.

9.5.4. 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% от 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.

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

Ответ: _____.

9.5.5. Документ объёмом 5 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^{28} бит в секунду;
- объём сжатого архиватором документа равен 60% от исходного;
- время, требуемое на сжатие документа, — 40 секунд, на распаковку — 10 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.

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

Ответ: _____.

10

ВЫЧИСЛЕНИЕ КОЛИЧЕСТВА ИНФОРМАЦИИ. ВЫЧИСЛЕНИЕ КОЛИЧЕСТВА ВАРИАНТОВ СООБЩЕНИЯ

10.1. Для кодирования различных сообщений используют флажки четырёх видов (красный, жёлтый, зелёный и синий). Сколько различных сообщений можно закодировать, если использовать в сообщении ровно три флажка в определённом порядке?

Решение.

Это классическая задача «поставить в ряд K объектов S различных видов». Количество способов это сделать равно S^K . Здесь $S = 4$, $K = 3$. Подставляем в формулу: $4^3 = 64$.

Ответ: 64.

10.1.1. Петя и Вася передают друг другу сообщения, используя двухцветный фонарик. На фонарике имеются две кнопки, соответствующие цветам, которыми будет светить фонарик (красный и синий). Две кнопки одновременно нажать нельзя. Для передачи сообщения мальчики последовательно нажимают шесть раз на какую-то кнопку. Сколько различных сообщений могут передать мальчики?

Ответ: _____.

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

Ответ: _____.

10.1.3. Для кодирования 300 различных сообщений используют 5 последовательных цветовых вспышек. Лампочки скольких различных цветов должны использоваться при передаче?

Ответ: _____.

10.1.4. Для кодирования 500 различных сообщений используют 4 последовательных цветовых вспышки. Лампочки скольких различных цветов должны использоваться при передаче?

Ответ: _____.

10.1.5. Петя и Вася передают друг другу сообщения, используя трёхцветный фонарик. На фонарике имеются три кнопки, соответствующие цветам, которыми будет светить фонарик (красный, зелёный и синий). Две кнопки одновременно нажать нельзя. Для передачи сообщения мальчики последовательно нажимают пять раз на какую-то кнопку. Сколько различных сообщений могут передать мальчики?

Ответ: _____.

10.2. Вася и Петя передают друг другу сообщения, используя жёлтый и синий фонарики. Эти они делают, включая фонарики последовательно на одинаково короткое время в некоторой комбинации. Количество вспышек в одном сообщении — 3 или 4. Сколько различных сообщений могут передавать мальчики?

Решение. Важно понимать, что при таком способе передачи информации используется не два различных сигнала (жёлтый и синий), а немного больше. Пауза между сообщениями позволяет определять, сколько в сообщении вспышек (3 или 4), и отличать сообщение с тремя вспышками от сообщения с четырьмя вспышками. Для каждого вида сообщений посчитаем количество различных вариантов по формуле S^K . Здесь S — количество различных символов в сообщении (2 — количество фонариков), K — количество символов в сообщении (количество вспышек — 3 или 4). Вычисляем: для трёх вспышек: $2^3 = 8$, для четырёх вспышек: $2^4 = 16$. Всего различных элементов: $8 + 16 = 24$.

Ответ: 24.

10.2.1. Азбука Морзе позволяет кодировать символы для сообщений по радиосвязи, задавая комбинацию точек и тире. Сколько различных символов (цифр, букв, знаков пунктуации и т. д.) можно закодировать, используя код Морзе длиной не менее трёх и не более четырёх сигналов (точек и тире)?

Ответ: _____.

10.2.2. Вася и Петя передают друг другу сообщения, используя синий и красный фонарики. Они делают это, включая фонарики последовательно на одинаково короткое время в некоторой комбинации. Количество вспышек в одном сообщении — 6 или 7. Сколько различных сообщений могут передавать мальчики?

Ответ: _____.

10.2.3. 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: _____.

10.2.4. Вася и Петя передают друг другу сообщения, используя флажки трёх цветов (синий, зелёный и красный). Для этого они поднимают последовательно один из флажков, потом опускают его и поднимают ещё какой-то. Наименьшее количество поднятий флажков в сообщении — 3, наибольшее — 5. Сколько различных сообщений могут передавать мальчики?

Ответ: _____.

10.2.5. Вася и Петя передают друг другу сообщения, используя синий, красный и зелёный фонарики. Они делают это, включая фонарики на одинаково короткое время в некоторой комбинации. Количество вспышек в одном сообщении — 2 или 4. Сколько различных сообщений могут передавать мальчики?

Ответ: _____.

10.3. Все 5-буквенные слова, составленные из букв А, Е, И, записаны в алфавитном порядке. Вот начало списка:

1. ААААА
2. ААААЕ
3. ААААИ
4. АААЕА

Запишите слово, стоящее на 102-м месте от начала списка.

Решение. Анализируем указанный способ порождения списка слов. Понимаем, что это — позиционная система счисления.

В ней используется всего 3 символа. Поэтому это — троичная система счисления. Обозначим используемые символы привычными цифрами (от 0 до 2). При этом соблюдаем алфавитный порядок: А—0, Е—1, И—2.

Запишем в этих обозначениях число, стоящее в первой строке: 1. 0000. Это число 0.

Во второй строке: 0001. Это число 1. В третьей строке: 0002. Это число 2. То есть, в строке номер 102 будет стоять число 101. Переведем это число в троичную систему счисления (будем делить на 3, пока не получим в результате деления ноль, и собирать в обратную сторону остатки от деления).

$101 : 3 = 33$, остаток 2,

$33 : 3 = 11$, остаток 0,

$11 : 3 = 3$, остаток 2,

$3 : 3 = 1$, остаток 0,

$1 : 3 = 0$, остаток 1.

Собираем остатки в обратном порядке: 10202₃. Заменяем каждую полученную цифру соответствующей ей буквой.

Ответ: ЕАИИИ.

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

1. АААА

2. АААЕ

3. АААИ

4. АААО

5. ААЕА

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

Ответ: _____.

10.3.2. Все 4-буквенные слова, составленные из букв Б, В, Г, Д, записаны в алфавитном порядке. Вот начало списка:

1. ББББ

2. БББВ

3. БББГ

4. БББД

5. ББВБ

Запишите слово, стоящее на 244-м месте от начала списка.

Ответ: _____.

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

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

Под каким номером в списке будет стоять слово ИОАЕ?

Ответ: _____.

10.3.4. Все 6-буквенные слова, составленные из букв А, Е, И, О, У, записаны в алфавитном порядке и пронумерованы. Вот начало списка:

1. АААААА
2. АААААЕ
3. АААААИ
4. АААААО
5. АААААУ
6. ААААЕА

Под каким номером в списке будет стоять слово ААИЕУО?

Ответ: _____.

10.3.5. Все 5-буквенные слова, составленные из букв С, Т, У, записаны в алфавитном порядке. Вот начало списка:

1. ССССС
2. ССССТ
3. ССССУ
4. СССТС
5. СССТТ

Запишите слово, стоящее на 127-м месте от начала списка.

Ответ: _____.

10.4. Вася и Петя передают друг другу сообщения, используя флажки шести цветов. Эти они делают, поднимая последовательно один из флажков, потом опускают его и поднимают ещё какой-то. Чтобы не путаться, ребята договорились, что флажки в одном сообщении не могут повторяться, при этом сообщение состоит из трёх флажков. Сколько различных сообщений могут передавать мальчики?

Решение.

Сообщение состоит из трёх флажков. Будем последовательно анализировать все три позиции, которым можно «задать» флажок определённого цвета. На первую позицию можно выбрать любой из шести флажков. Так как флажки в сообщении не могут повторяться, на вторую позицию можно будет выбрать любой из пяти оставшихся флажков. А, значит, на третью позицию можно будет выбрать только любой из четырёх оставшихся флажков. Ответ получаем умножением этих чисел: $6 \cdot 5 \cdot 4 = 120$.

Ответ: 120.

10.4.1. Вася и Петя передают друг другу сообщения, используя 5 флажков. Все флажки различных цветов. Они передают сообщение, поднимая последовательно один из флажков, потом опускают его и поднимают ещё какой-то. Чтобы не путаться, ребята договорились, что флажки в одном сообщении не могут повторяться, при этом в каждом сообщении мальчики используют все имеющиеся флажки. Сколько различных сообщений могут передавать мальчики?

Ответ: _____.

10.4.2. 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: _____.

10.4.3. В некотором племени для общения используются всего 64 слова. Вождь племени на общем собрании произносит 32 слова. Какое количество информации (число бит) сообщает вождь?

Ответ: _____.

10.4.4. Вася и Петя передают друг другу сообщения, используя флажки шести цветов. Они делают это, поднимая последовательно один из флажков, потом опускают его и поднимают ещё какой-то. Чтобы не путаться, ребята договорились, что флажки в одном сообщении не могут повторяться, при этом сообщение состоит из трёх флажков. Сколько различных сообщений могут передавать мальчики?

Ответ: _____.

10.4.5. Вася и Петя передают друг другу сообщения, используя флажки семи цветов. Они делают это, поднимая последовательно один из флажков, потом опускают его и поднимают ещё какой-то. Чтобы не путаться, ребята договорились, что флажки в одном сообщении не могут повторяться, при этом сообщение состоит из четырёх флажков. Сколько различных сообщений могут передавать мальчики?

Ответ: _____.

10.5. Вася составляет 4-буквенные слова, в которых есть только буквы В, Л, А, С, Я, причём в каждом слове только одна гласная буква. 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?

Решение. Сначала проанализируем, сколькими способами можно поставить в слове гласную букву. Используется только одна из двух гласных букв. Это 2 варианта. Выбранную гласную букву можно поставить на любую из четырёх возможных позиций в слове. Значит, это ещё 4 варианта. На оставшихся трёх позициях можно поставить любую из согласных букв (их 3). Это ещё 3^3 вариантов. Все эти варианты выбираются независимо друг от друга, поэтому их перемножаем: $2 \cdot 4 \cdot 3^3 = 8 \cdot 27 = 216$.

Ответ: 216.

10.5.1. Вася составляет 5-буквенные слова, в которых есть только буквы: С, Т, Ё, П, А, — причём в каждом слове есть только одна гласная буква, которая стоит либо на первом, либо на последнем месте. 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: _____.

10.5.2. Вася составляет 4-буквенные слова, в которых есть только буквы: С, Т, Ё, П, А, — причём буква Ё 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: _____.

10.5.3. 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: _____.

10.5.4. 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: _____.

10.5.5. Вася составляет 4-буквенные слова, в которых есть только буквы: С, Т, Ё, П, А, — причём гласная буква (А или Ё) используется в каждом слове ровно 1 раз и стоит на первом или на последнем месте. 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: _____.

11.1. 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

$F(n) = F(n-2) + 3$, при $n > 1$ и n не кратно 2

$F(1) = 1$

Чему равно значение функции $F(8)$?

В ответе запишите только натуральное число.

Решение. Будем вычислять значение функции последовательно, начиная с первого неизвестного элемента. Известный элемент имеется только $F(1)$. Будем вычислять элементы, начиная с $F(2)$.

$F(2) = F(2-1) + F(2/2) = F(1) + F(1) = 1 + 1 = 2$ ($n = 2$, кратно 2),

$F(3) = F(3-2) + 3 = F(1) + 3 = 1 + 3 = 4$ ($n = 3$, не кратно 2),

$F(4) = F(4-1) + F(4/2) = F(3) + F(2) = 4 + 2 = 6$ ($n = 4$, кратно 2),

$F(5) = F(5-2) + 3 = F(3) + 3 = 4 + 3 = 7$ ($n = 5$, не кратно 2),

$F(6) = F(6-1) + F(6/2) = F(5) + F(3) = 7 + 4 = 11$ ($n = 6$, кратно 2),

$F(7) = F(7-2) + 3 = F(5) + 3 = 7 + 3 = 10$ ($n = 7$, не кратно 2),

$F(8) = F(8-1) + F(8/2) = F(7) + F(4) = 10 + 6 = 16$ ($n = 8$, кратно 2),

Ответ: 16.

Заметим, что вычисление $F(8)$ можно сделать и напрямую, начиная с $F(8)$. Но при этом придётся несколько раз приостанавливать вычисление очередного элемента и начинать вычислять другой, значение которого ещё не известно. А потом возвращаться обратно. Например, $F(8) = F(7) + F(4)$. оба значения ($F(7)$ и $F(4)$) пока не известны. Нужно приостановить вычисление $F(8)$ и начать вычисление $F(7)$, а потом и $F(4)$, которые, в свою очередь, тоже придётся приостанавливать, чтобы вычислить ещё неизвестные значения элементов, от которых зависят $F(7)$ и $F(4)$. И так до тех пор, пока вычисления не приведут к известному значению $F(1)$. Потом нужно будет возвращаться во все приостановленные вычисления, подставляя недостающие значения. Такой способ неудобен и более громоздок. Мы не рекомендуем его использовать. Однако для проверки правильности ответа, возможно, имеет смысл использовать именно такой, альтернативный метод решения.

Другой способ оформления вычислений. Будем записывать все вычисления в таблицу, последовательно вычисляя значения элементов от $F(2)$ до $F(8)$:

n	1	2	3	4
F(n)	1	2	4	6
Пояснение	Нач. знач-е	$F(1)+F(1) = 1+1$	$F(1)+3 = 1+3$	$F(3)+F(2) = 4+2$

Окончание табл.

n	5	6	7	8
F(n)	7	11	10	16
Пояснение	$F(3)+3 = 4+3$	$F(5)+F(3) = 7+4$	$F(5)+3 = 7+3$	$F(7)+F(4) = 10+6$

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

$$F(n) = F(n-1) + n - 2, \text{ при } n > 1$$

$$F(1) = 2$$

Чему равно значение функции $F(7)$?

В ответе запишите только натуральное число.

Ответ: _____.

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

$$F(n) = F(n-1) + n/2, \text{ при } n > 1 \text{ и } n \text{ кратно } 2$$

$$F(n) = F(n-1) + 1, \text{ при } n > 1 \text{ и } n \text{ не кратно } 2$$

$$F(1) = 1$$

Чему равно значение функции $F(8)$?

В ответе запишите только натуральное число.

Ответ: _____.

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

$$F(n) = F(n-1) + F(n/2), \text{ при } n > 1 \text{ и } n \text{ кратно } 2$$

$$F(n) = F(n-1) + 1, \text{ при } n > 1 \text{ и } n \text{ не кратно } 2$$

$$F(1) = 1$$

Чему равно значение функции $F(8)$?

В ответе запишите только натуральное число.

Ответ: _____.

11.1.4. 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: _____.

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

$F(n) = F(n - 1) + F(n/3)$, при $n > 1$ и n кратно 3

$F(n) = F(n - 1) + 1$, при $n > 1$ и n не кратно 3

$F(1) = 2$

Чему равно значение функции $F(12)$?

В ответе запишите только натуральное число.

Ответ: _____.

11.2. Определите, сколько звёздочек будет напечатано в результате вызова $F(6)$ приведённой подпрограммы:

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач вывод '*' если n > 2 то F(div(n, 2)) F(n - 1) все кон </pre>	<pre> procedure F(n: integer); begin write('*'); if n > 2 then begin F(n div 2); F(n - 1) end end; end; </pre>
C++	Бейсик
<pre> void F(int n) { std::cout << "*"; if (n > 2) { F(n / 2); F(n - 1); } } </pre>	<pre> SUB F(n) PRINT "*"; IF n > 2 THEN F(n \ 2) F(n - 1) END IF END SUB </pre>

Python

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

Решение. Обозначим $R(n)$ — количество звёздочек, которые выводит на экран подпрограмма $F(n)$. Анализируя подпрограмму, составим рекуррентное уравнение для вычисления $R(n)$. Заметим, что при невыполнении условия « $n > 2$ » рекуррентные вызовы ($F(n \text{ div } 2)$ и $F(n-1)$) не происходят — вывод звёздочки находится вне блока условия и, значит, выполняется независимо от выполнения условия. То есть, при « $n \leq 2$ » подпрограмма просто выводит на экран одну звёздочку и заканчивает свою работу. Значит, можем записать, что « $R(n) = 1$ при $n \leq 2$ » (выводит одну (1)) звёздочку. А при выполнении условия « $n > 2$ » подпрограмма не только выводит одну звёздочку, но и вызывает $F(n \text{ div } 2)$ и $F(n-1)$. Которые, в свою очередь, тоже выводят своё количество звёздочек. По нашим обозначениям, $F(n \text{ div } 2)$ выводит $R(n \text{ div } 2)$ звёздочек. И аналогично $F(n-1)$ выводит $R(n-1)$ звёздочек.

Получаем выражение: « $R(n) = 1 + R(n \text{ div } 2) + R(n-1)$, при $n > 2$ ». Запишем всё вместе в удобном виде:

$R(n) = R(n \text{ div } 2) + R(n-1) + 1$, при $n > 2$,

$R(n) = 1$, при $n \leq 2$.

Вычислим последовательно $R(n)$ до $R(6)$. Определим, начиная с какого элемента нужно вычислять $R(n)$. Рекуррентные вызовы $R(n \text{ div } 2)$ и $R(n-1)$ происходят при $n > 2$. То есть, наименьшее значение n , вызывающее рекурсию, будет $n = 3$. Значит, начать вычисления нужно будет с $R(3 \text{ div } 2) = R(1)$ и $R(3-1) = R(2)$.

Будем вычислять последовательно $R(n)$, начиная с $R(1)$:

$R(1) = 1$ ($n = 1 \leq 2$),

$R(2) = 1$ ($n = 2 \leq 2$),

$R(3) = R(3 \text{ div } 2) + R(3-1) + 1 = R(1) + R(2) + 1 = 1 + 1 + 1 = 3$
($n = 3 \geq 2$),

$R(4) = R(4 \text{ div } 2) + R(4-1) + 1 = R(2) + R(3) + 1 = 1 + 3 + 1 = 5$
($n = 4 \geq 2$),

$R(5) = R(5 \text{ div } 2) + R(5-1) + 1 = R(2) + R(4) + 1 = 1 + 5 + 1 = 7$
($n = 5 \geq 2$),

$R(6) = R(6 \text{ div } 2) + R(6-1) + 1 = R(3) + R(5) + 1 = 3 + 7 + 1 = 11$
($n = 6 \geq 2$).

Ответ: 11.

11.2.1. Определите, сколько звёздочек будет напечатано в результате вызова $F(5)$ приведённой подпрограммы:

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 1 то F(div(n, 2)) F(n - 1) все вывод "*" кон </pre>	<pre> procedure F(n: integer); begin if n > 1 then begin F(n div 2); F(n - 1) end; write('*') end; </pre>
C++	Бейсик
<pre> void F(int n) { if (n > 1) { F(n / 2); F(n - 1); } std::cout << "*"; } </pre>	<pre> SUB F(n) IF n > 1 THEN F(n \ 2) F(n - 1) END IF PRINT "*"; END SUB </pre>
Python	
<pre> def F(n): if n > 1: F(n // 2) F(n - 1) print("*") </pre>	

Ответ: _____.

11.2.2. Определите, сколько звёздочек будет напечатано в результате вызова $F(3)$ приведённой подпрограммы:

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 0 то F(n - 2) F(n - 1) F(n - 1) все вывод "*" кон </pre>	<pre> procedure F(n: integer); begin if n > 0 then begin F(n - 2); F(n - 1); F(n - 1) end; write('*'); end; </pre>

C++	Бейсик
<pre>void F(int n) { if (n > 0) { F(n - 2); F(n - 1); F(n - 1); } std::cout << "*"; }</pre>	<pre>SUB F(n) IF n > 0 THEN F(n - 2) F(n - 1) F(n - 1) END IF PRINT "*"; END SUB</pre>
Python	
<pre>def F(n): if n > 0: F(n - 2) F(n - 1) F(n - 1) print("*")</pre>	

Ответ: _____.

11.2.3. Определите, сколько звёздочек будет напечатано в результате вызова $F(7)$ приведённой подпрограммы:

Алгоритмический язык	Паскаль
<pre><u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>если</u> n > 1 <u>то</u> <u>вывод</u> "*" F(n - 1) F(div(n, 2)) <u>все</u> <u>кон</u></pre>	<pre>procedure F(n: integer); begin if n > 1 then begin write('*'); F(n - 1); F(n div 2) end end; end;</pre>
C++	Бейсик
<pre>void F(int n) { if (n > 1) { std::cout << "*"; F(n - 1); F(n / 2); } }</pre>	<pre>SUB F(n) IF n > 1 THEN PRINT "*"; F(n - 1) F(n \ 2) END IF END SUB</pre>

Python
<pre>def F(n): if n > 1: print("*") F(n - 1) F(n // 2)</pre>

Ответ: _____.

11.2.4. Определите, сколько звёздочек будет напечатано в результате вызова F(6) приведённой подпрограммы:

Алгоритмический язык	Паскаль
<u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>вывод</u> "*" <u>если</u> n > 2 <u>то</u> F(n - 2) F(n - 1) <u>все</u> <u>кон</u>	procedure F(n: integer); begin write('*'); if n > 2 then begin F(n - 2); F(n - 1) end end;
C++	Бейсик
void F(int n) { std::cout << "*"; if (n > 2) { F(n - 2); F(n - 1); } }	SUB F(n) PRINT "*"; IF n > 2 THEN F(n - 2) F(n - 1) END IF END SUB
Python	
def F(n): print("*") if n > 2: F(n - 2) F(n - 1)	

Ответ: _____.

11.2.5. Определите, сколько звёздочек будет напечатано в результате вызова **F(1)** приведённой подпрограммы:

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач вывод "*" если n < 6 то F(n + 2) F(n + 1) все кон </pre>	<pre> procedure F(n: integer); begin write('*'); if n < 6 then begin F(n + 2); F(n + 1) end end; </pre>
C++	Бейсик
<pre> void F(int n) { std::cout << "*"; if (n < 6) { F(n + 2); F(n + 1); } } </pre>	<pre> SUB F(n) PRINT "*"; IF n < 6 THEN F(n + 2) F(n + 1) END IF END SUB </pre>
Python	
<pre> def F(n): print("*") if n < 6: F(n + 2) F(n + 1) </pre>	

Ответ: _____.

11.3. Даны рекурсивные алгоритмы F и G. Чему равно значение функции F(5)?

Алгоритмический язык	Паскаль
<pre> алг <u>цел</u> F(<u>цел</u> n) нач если n > 1 то <u>знач</u> := F(n-2) + G(n-1) иначе <u>знач</u> := n все кон алг <u>цел</u> G(<u>цел</u> n) нач если n > 2 то <u>знач</u> := G(n-1) + F(n-1) иначе <u>знач</u> := n все кон </pre>	<pre> function F(n:integer):integer; begin if n > 1 then F := F(n - 2) + G(n - 1) else F := n end; function G(n:integer):integer; begin if n > 2 then G := G(n - 1) + F(n - 1) else G := n end; </pre>
C++	Бейсик
<pre> int F(int n) { if (n > 1) return F(n-2)+G(n-1); else return n; } int G(int n) { if (n > 2) return G(n-1)+F(n-1); else return n; } </pre>	<pre> FUNCTION F(n) IF n > 1 THEN F = F(n - 2) + G(n - 1) ELSE F = n END IF END FUNCTION FUNCTION G(n) IF n > 2 THEN G = G(n - 1) + F(n - 1) ELSE G = n END IF END FUNCTION </pre>
Python	
<pre> def F(n): if n > 1: return F(n - 2) + G(n - 1) else: return n def G(n): if n > 2: return G(n - 1) + F(n - 1) else: return n </pre>	

Решение. В приведённой программе непосредственно содержатся рекуррентные формулы для вычисления $F(n)$ и $G(n)$. А именно:

$$F(n) = F(n-2) + G(n-1), \text{ при } n > 1,$$

$$F(n) = n, \text{ при } n \leq 1.$$

$$G(n) = G(n-1) + F(n-1), \text{ при } n > 2,$$

$$G(n) = n, \text{ при } n \leq 2.$$

Будем последовательно вычислять $F(n)$ и $G(n)$, начиная с маленьких n , до $n = 5$. Найдём начальное требуемое n . Для $F(n)$ требуется вычислять $F(n-2)$ и $G(n-1)$ для наименьшего $n = 2$.

То есть, $F(2-2) = F(0)$ и $G(2-1) = G(1)$. Для $G(n)$ требуется вычислять $G(n-1)$ и $F(n-1)$ для наименьшего $n = 3$. То есть, $G(3-1) = G(2)$ и $F(3-1) = F(2)$. Получаем, что наименьшие значения n требуются для $F(0)$ и $G(1)$. Вычисляем последовательно $F(n)$ и $G(n)$:

$$F(0) = 0 \quad (n = 0 \leq 1),$$

$$G(1) = 1 \quad (n = 1 \leq 2),$$

$$F(1) = 1 \quad (n = 1 \leq 1),$$

$$G(2) = 2 \quad (n = 2 \leq 2),$$

$$F(2) = F(2-2) + G(2-1) = F(0) + G(1) = 0 + 1 = 1 \quad (n = 2 > 1),$$

$$G(3) = G(3-1) + F(3-1) = G(2) + F(2) = 2 + 1 = 3 \quad (n = 3 > 2),$$

$$F(3) = F(3-2) + G(3-1) = F(1) + G(2) = 1 + 2 = 3 \quad (n = 3 > 1),$$

$$G(4) = G(4-1) + F(4-1) = G(3) + F(3) = 3 + 3 = 6 \quad (n = 4 > 2),$$

$$F(4) = F(4-2) + G(4-1) = F(2) + G(3) = 1 + 3 = 4 \quad (n = 4 > 1),$$

$$G(5) = G(5-1) + F(5-1) = G(4) + F(4) = 6 + 4 = 10 \quad (n = 5 > 2),$$

$$F(5) = F(5-2) + G(5-1) = F(3) + G(4) = 3 + 6 = 9 \quad (n = 5 > 1),$$

Ответ: 9.

Другой способ оформления решения. Для удобства можно оформить те же записи при помощи таблицы, в которой записывать все вычисляемые значения для $F(n)$ и $G(n)$:

n	0	1	2	3	4	5
F(n)	0	1	1	3	4	9
Пояснение	Нач. знач-е	Нач. знач-е	$F(0)+G(1) = 0+1$	$F(1)+G(2) = 1+2$	$F(2)+G(3) = 1+3$	$F(3)+G(4) = 3+6$
G(n)	0	1	2	3	6	10
Пояснение	Нач. знач-е	Нач. знач-е	Нач. знач-е	$G(2)+F(2) = 2+1$	$G(3)+F(3) = 3+3$	$G(4)+F(4) = 6+4$

Пояснения к вычислениям можно не писать. Но по ним после можно проверить вычисления.

Ответ: 9.

11.3.1. Даны рекурсивные алгоритмы F и G. Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(20)?

Алгоритмический язык	Паскаль
<pre> 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 </pre>	<pre> 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; end; </pre>
C++	Бейсик
<pre> 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); } </pre>	<pre> 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 </pre>
Python	
<pre> def F(n): if n > 0: G(n - 1) def G(n): print("*") if n > 1: F(n - 2) </pre>	

Ответ: _____.

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

Алгоритмический язык	Паскаль
<pre> алг <u>цел</u> F(<u>цел</u> n) нач если n>1 то <u>знач</u>:=F(n-1)+G(n-1) иначе <u>знач</u>:=n все кон алг <u>цел</u> G(<u>цел</u> n) нач если n>2 то <u>знач</u>:=G(n-1)+F(n) иначе <u>знач</u>:=n все кон </pre>	<pre> 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; </pre>
C++	Бейсик
<pre> 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; } </pre>	<pre> 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 </pre>

Python
<pre>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</pre>

Ответ: _____.

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

Алгоритмический язык	Паскаль
<pre><u>алг</u> <u>цел</u> F(<u>цел</u> n) <u>нач</u> <u>если</u> n > 2 <u>то</u> <u>знач</u>:=F(n-1)+G(n-2) <u>иначе</u> <u>знач</u>:=n <u>все</u> <u>кон</u> <u>алг</u> <u>цел</u> G(<u>цел</u> n) <u>нач</u> <u>если</u> n > 1 <u>то</u> <u>знач</u>:=G(n-1)+F(n) <u>иначе</u> <u>знач</u>:=n+1 <u>все</u> <u>кон</u></pre>	<pre>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;</pre>

C++	Бейсик
<pre> 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; } </pre>	<pre> 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 </pre>
Python	
<pre> 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 </pre>	

Ответ: _____.

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

Алгоритмический язык	Паскаль
<pre> алг <u>цел</u> F(<u>цел</u> n) нач если n > 2 то <u>знач</u>:=F(n-1)+G(n-2) иначе <u>знач</u>:=3-n все кон алг <u>цел</u> G(<u>цел</u> n) нач если n > 2 то <u>знач</u>:=G(n-1)+F(n-1) иначе <u>знач</u>:=n+1 все <u>кон</u> </pre>	<pre> function F(n:integer):integer; begin if n > 2 then F:=F(n-1)+G(n-2) else F:=3-n end; function G(n:integer):integer; begin if n > 2 then G:=G(n-1)+F(n-1) else G:=n+1 end; </pre>
C++	Бейсик
<pre> int F(int n) { if (n > 2) return F(n-1)+G(n-2); else return 3-n; } int G(int n) { if (n > 2) return G(n-1)+F(n-1); else return n + 1; } </pre>	<pre> FUNCTION F(n) IF n > 2 THEN F=F(n-1)+G(n-2) ELSE F=3-n END IF END FUNCTION FUNCTION G(n) IF n > 2 THEN G=G(n-1)+F(n-1) ELSE G=n+1 END IF END FUNCTION </pre>

Python
<pre>def F(n): if n > 2: return F(n - 1) + G(n - 2) else: return 3 - n def G(n): if n > 2: return G(n - 1) + F(n - 1) else: return n + 1</pre>

Ответ: _____.

11.3.5. Даны рекурсивные алгоритмы F и G. Чему равна сумма всех чисел, напечатанных на экране при выполнении вызова F(15)?

Алгоритмический язык	Паскаль
<pre>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</pre>	<pre>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;</pre>
C++	Бейсик
<pre>void G(int n); void F(int n) { if (n > 0) G(n - 2); } void G(int n) { std::cout << n; if (n > 1) F(n - 1); }</pre>	<pre>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</pre>

Python

```
def F(n):
    if n > 0:
        G(n - 2)
def G(n):
    print(n)
    if n > 1:
        F(n - 1)
```

Ответ: _____.

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

Алгоритмический язык	Паскаль
<u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>ВЫВОД</u> n <u>если</u> n > 1 <u>то</u> F(div(n,2)) F(n - 2) <u>все</u> <u>кон</u>	procedure F(n: integer); begin write(n); if n > 1 then begin F(n div 2); F(n - 2) end end; end;
C++	Бейсик
<pre>void F(int n) { std::cout << n; if (n > 1) { F(n / 2); F(n - 2); } }</pre>	<pre>SUB F(n) PRINT n; IF n > 1 THEN F(n / 2) F(n - 2) END IF END SUB</pre>
Python	
<pre>def F(n): print(n) if n > 1: F(n // 2) F(n - 2)</pre>	

Решение.

Выполним вызовы $F(n)$ последовательно, начиная с начального значения n до $F(5)$.

Определим начальное значение n .

Рекуррентные вызовы $F(n/2)$ и $F(n-2)$ делаются для наименьшего $n > 1$, т.е., для $n = 2$ выполняются $F(2/2) = F(1)$ и $F(2-2) = F(0)$. Значит, начнем 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 $F(2/2) = F(1)$ 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	311	Условие $n > 1$ снова выполняется. Выводим на экран 3, а потом вызываем $F(3/2) = F(1)$ и $F(3-2) = F(1)$. Дописываем их результаты справа от цифры 3.
4	4210210	Условие выполняется. Выводим 4. Вызываем $F(4/2) = F(2)$ и $F(4-2) = F(2)$. Дописываем их результаты справа от 4.
5	5210311	Условие выполняется. Выводим 5. Вызываем $F(5/2) = F(2)$ и $F(5-2) = F(3)$. Дописываем их результаты справа от 5.

Ответ: 5210311.

11.4.1. Дан рекурсивный алгоритм F . Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова $F(1)$.

Алгоритмический язык	Паскаль
<u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>вывод</u> n <u>если</u> n < 3 <u>то</u> F(n * 2) F(n + 2) <u>все</u> <u>кон</u>	procedure F(n: integer); begin write(n); if n < 3 then begin F(n * 2); F(n + 2) end end; end;
C++	Бейсик
void F(int n) { std::cout << n; if (n < 3) { F(n * 2); F(n + 2); } } }	SUB F(n) PRINT n; IF n < 3 THEN F(n * 2) F(n + 2) END IF END SUB
Python	
def F(n): print(n) if n < 3: F(n * 2) F(n + 2)	

Ответ: _____.

11.4.2. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(1).

Алгоритмический язык	Паскаль
<u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>вывод</u> n <u>если</u> n < 4 <u>то</u> F(n + 2) F(n + 1) <u>все</u> <u>кон</u>	procedure F(n: integer); begin write(n); if n < 4 then begin F(n + 2); F(n + 1) end end; end;

C++	Бейсик
<pre>void F(int n) { std::cout << n; if (n < 4) { F(n + 2); F(n + 1); } }</pre>	<pre>SUB F(n) PRINT n; IF n < 4 THEN F(n + 2) F(n + 1) END IF END SUB</pre>
Python	
<pre>def F(n): print(n) if n < 4: F(n + 2) F(n + 1)</pre>	

Ответ: _____.

11.4.3. Дан рекурсивный алгоритм F.

Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(4).

Алгоритмический язык	Паскаль
<pre><u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>вывод</u> n <u>если</u> n > 1 <u>то</u> F(n - 1) F(n - 2) <u>все</u> <u>кон</u></pre>	<pre>procedure F(n: integer); begin write(n); if n > 1 then begin F(n - 1); F(n - 2) end end; end;</pre>
C++	Бейсик
<pre>void F(int n) { std::cout << n; if (n > 1) { F(n - 1); F(n - 2); } }</pre>	<pre>SUB F(n) PRINT n; IF n > 1 THEN F(n - 1) F(n - 2) END IF END SUB</pre>

Python
<pre>def F(n): print(n) if n > 1: F(n - 1) F(n - 2)</pre>

Ответ: _____.

11.4.4. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(4).

Алгоритмический язык	Паскаль
<pre>алг F(цел n) нач вывод n если n > 1 то F(div(n, 2)) F(n - 1) все кон</pre>	<pre>procedure F(n: integer); begin write(n); if n > 1 then begin F(n div 2); F(n - 1) end end; end;</pre>
C++	Бейсик
<pre>void F(int n) { std::cout << n; if (n > 1) { F(n / 2); F(n - 1); } }</pre>	<pre>SUB F(n) PRINT n; IF n > 1 THEN F(n \ 2) F(n - 1) END IF END SUB</pre>
Python	
<pre>def F(n): print(n) if n > 1: F(n // 2) F(n - 1)</pre>	

Ответ: _____.

11.4.5. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(4).

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 1 то F(div(n, 2)) F(n - 1) все вывод n кон </pre>	<pre> procedure F(n: integer); begin if n > 1 then begin F(n div 2); F(n - 1) end; write(n) end; </pre>
C++	Бейсик
<pre> void F(int n) { if (n > 1) { F(n / 2); F(n - 1); } std::cout << n; } </pre>	<pre> SUB F(n) IF n > 1 THEN F(n \ 2) F(n - 1) END IF PRINT n; END SUB </pre>
Python	
<pre> def F(n): if n > 1: F(n // 2) F(n - 1) print(n) </pre>	

Ответ: _____.

Рекомендации к решению заданий 11.5. Рекомендуем построить таблицу для последовательного вычисления последовательности чисел, выводимой на экран подпрограммой. Обращайте внимание на положение в подпрограмме оператора вывода на экран относительно проверяемого условия. Если вывод на экран выполняется только при выполнении условия, то при ложном условии подпрограмма ничего на экран не выводит. Важно, в каком месте стоит вывод на экран относительно рекуррентных вызовов. Если вывод на экран записан между вызовов рекурсии, то и записывать результат вывода нужно будет после результата первой рекурсии и перед результатом второго вызова. А если вывод на экран записан после рекуррентных вызовов, то следует сначала записывать друг за другом их результаты, а потом уже (справа) записывать результат вывода на экран.

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

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 1 то F(div(n, 2)) вывод n F(n - 1) все кон </pre>	<pre> procedure F(n: integer); begin if n > 1 then begin F(n div 2); write(n); F(n - 1) end end; </pre>
C++	Бейсик
<pre> void F(int n) { if (n > 1) { F(n / 2); std::cout << n; F(n - 1); } } </pre>	<pre> SUB F(n) IF n > 1 THEN F(n \ 2) PRINT n; F(n - 1) END IF END SUB </pre>
Python	
<pre> def F(n): if n > 1: F(n // 2) print(n) F(n - 1) </pre>	

Ответ: _____.

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

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 0 то вывод n F(div(n, 3)) F(n - 1) все кон </pre>	<pre> procedure F(n: integer); begin if n > 0 then begin write(n); F(n div 3); F(n - 1) end end; </pre>

С++	Бейсик
<pre>void F(int n) { if (n > 0) { std::cout << n; F(n / 3); F(n - 1); } }</pre>	<pre>SUB F(n) IF n > 0 THEN PRINT n; F(n \ 3) F(n - 1) END IF END SUB</pre>
Python	
<pre>def F(n): if n > 0: print(n) F(n // 3) F(n - 1)</pre>	

Ответ: _____.

11.5.3. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(4).

Алгоритмический язык	Паскаль
<pre><u>алг</u> F(<u>цел</u> n) <u>нач</u> <u>если</u> n > 1 <u>то</u> F(n - 1) F(div(n,2)) <u>вывод</u> n <u>все</u> <u>кон</u></pre>	<pre>procedure F(n: integer); begin if n > 1 then begin F(n - 1); F(n div 2); write(n) end end; end;</pre>
С++	Бейсик
<pre>void F(int n) { if (n > 1) { F(n - 1); F(n / 2); std::cout << n; } }</pre>	<pre>SUB F(n) IF n > 1 THEN F(n - 1) F(n \ 2) PRINT n; END IF END SUB</pre>

Python
<pre>def F(n): if n > 1: F(n - 1) F(n // 2) print(n)</pre>

Ответ: _____.

11.5.4. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(9, 0)?

Алгоритмический язык	Паскаль
<pre>алг F(цел n, k) нач если k = 2 то вывод n иначе F(div(n, 2), k + 1) F(n - 3, k + 1) все кон</pre>	<pre>procedure F(n, k: integer); begin if k = 2 then write(n) else begin F(n div 2, k + 1); F(n - 3, k + 1) end end;</pre>
C++	Бейсик
<pre>void F(int n, int k) { if (k == 2) std::cout << n; else { F(n / 2, k + 1); F(n - 3, k + 1); } }</pre>	<pre>SUB F(n, k) IF k = 2 THEN PRINT n ELSE F(n \ 2, k + 1) F(n - 3, k + 1) END IF END SUB</pre>
Python	
<pre>def F(n, k): if k == 2: print(n) else F(n // 2, k + 1) F(n - 3, k + 1)</pre>	

Ответ: _____.

11.5.5. Дан рекурсивный алгоритм F. Приведите последовательность чисел (без пробелов), напечатанных на экране при выполнении вызова F(1, 2)?

Алгоритмический язык	Паскаль
<pre> алг F(цел n, k) нач если k = 0 то вывод n иначе F(n + 2, k - 1) F(n + 4, k - 1) все кон </pre>	<pre> procedure F(n,k:integer); begin if k = 0 then write(n) else begin F(n + 2, k - 1); F(n + 4, k - 1) end end; </pre>
C++	Бейсик
<pre> void F(int n, int k) { if (k == 0) std::cout << n; else { F(n + 2, k - 1); F(n + 4, k - 1); } } </pre>	<pre> SUB F(n,k) IF k = 0 THEN PRINT n ELSE F(n + 2, k - 1) F(n + 4, k - 1) END IF END SUB </pre>
Python	
<pre> def F(n,k): if k == 0: print(n) else F(n + 2, k - 1) F(n + 4, k - 1) </pre>	

Ответ: _____.

Общая часть задания для задач 12.1.

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.

12.1. Сколько компьютеров может быть в сети с маской:
255.255.255.240?

Решение.

Переведём в двоичную систему счисления числа маски. Нас интересует только количество нулей в маске. Поэтому первые числа маски 255 ($= 1111111_2$) мы не будем рассматривать (в них нет нулей). В данном случае среди чисел маски есть только одно число, не равное 255. Это число 240. Переведём его в двоичную систему счисления. Получим 11110000_2 . В этом числе 4 нуля. Значит, для записи адреса узла в сети отведено только 4 двоичные цифры. При помощи четырёх двоичных цифр можно составить $2^4 = 16$ различных двоичных чисел. Но, как написано в условии, адрес узла не может состоять из одних только нулей и из одних только единиц. То есть, среди этих 16-ти чисел нас не устраивают 2. Остаётся 14 ($= 16 - 2$) различных вариантов адреса узла в сети.

Ответ: 14.

12.1.1. Сколько компьютеров может быть в сети с маской:
255.255.255.224?

Ответ: _____.

12.1.2. Сколько компьютеров может быть в сети с маской:
255.255.248.0?

Ответ: _____.

12.1.3. Сколько компьютеров может быть в сети с маской:
255.255.255.128?

Ответ: _____.

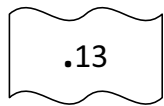
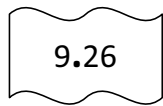

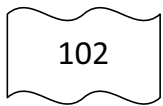
12.1.4. Сколько компьютеров может быть в сети с маской:
255.255.255.192?

Ответ: _____.

12.1.5. Сколько компьютеров может быть в сети с маской:
255.255.254.0?

Ответ: _____.

12.2.

			
A	B	C	D

Решение.

Будем использовать ограничения, которые накладываются на IP-адрес. То есть, IP-адрес записывается четырьмя числами, разделённых точками. Каждое число должно находиться в пределах от 0 до 255.

Анализируем имеющиеся фрагменты.

Замечаем, что на фрагменте D написано трёхзначное число без точки. Дописывать слева или справа к этому числу ещё одну цифру нельзя (получится число, большее 255). Значит, это целиком одно из четырёх чисел IP-адреса. И это число должно обязательно стоять в IP-адресе на первом месте.

Если бы перед этим числом находился какой-нибудь другой фрагмент IP-адреса, то он должен был бы заканчиваться точкой. А таких фрагментов в условии нет. Поставив фрагмент D на первое место и, учитывая, что это полное первое число IP-адреса, понимаем, что после этого числа должна стоять точка. Среди фрагментов есть только один, начинающийся с точки. Это фрагмент A. Получаем начальную часть IP-адреса: 102.13

Оставшиеся два фрагмента можно приставить к этим только двумя способами (сначала B, потом C, или наоборот):

102.139.265.6

102.135.69.26

Первый из этих вариантов не подходит, потому что третьим числом оказывается число 265, которого в IP-адресе быть не может.

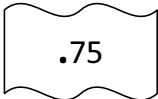
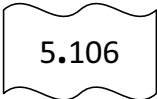
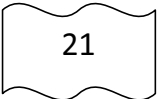

Остаётся единственный вариант: 102.135.69.26.

Запишем его обратно при помощи обозначений клочков бумаги, которые ему соответствуют.

Ответ: DACB.

12.2.1. 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.


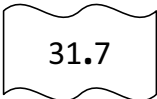

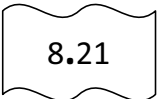
В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

			
А	Б	В	Г

Ответ: _____.

12.2.2. 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.

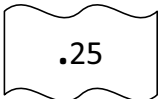
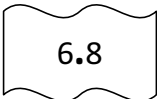

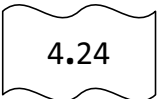
В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

			
А	Б	В	Г

Ответ: _____.

12.2.3. 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.

			
А	Б	В	Г

Ответ: _____.

12.2.4. 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.

3.231	3.25	18	.64
А	Б	В	Г

Ответ: _____.

12.2.5. 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.

9.69	25	1.21	.41
А	Б	В	Г

Ответ: _____.

12.3. 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: 137.219.167.47

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.

А	Б	С	Д	Е	Ф	Г	Н
0	47	137	160	167	219	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

Решение.

Формально для решения задачи нужно перевести каждую часть IP-адреса и каждую часть маски сети в двоичную систему счисления и записать в виде 8-разрядного числа (добавить слева нули до восьми разрядов). Затем получившиеся два 32-разрядных двоичных числа записать друг под другом и поразрядно перемножить каждую пару из 32-х разрядов. Затем разбить обратно 32-разрядный результат на группы по 8 разрядов и каждую группу как отдельное двоичное число перевести в десятичную систему счисления. Однако часть этих действий можно не делать, потому что результат получится очевидный.

Например, рассмотрим первые два числа IP-адреса и маски. Переведём каждое в двоичную систему счисления: $137 = 10001001_2$, $255 = 11111111_2$. Теперь запишем получившиеся двоичные числа друг под другом и поразрядно перемножим:

$$\begin{array}{r} 10001001 \\ \times 11111111 \\ \hline 10011001 \end{array}$$

Заметим, что так как маска состоит в двоичном виде из всех единиц (число 255), то при умножении разрядов IP-адреса на такую маску разряды IP-адреса остаются такими же, какими были. То есть, те фрагменты IP-адреса, которые расположены напротив чисел 255 маски, останутся неизменными. Значит, первая часть адреса сети будет 137.219.

Также заметим, что последняя часть маски сети равна нулю. При переводе в двоичную систему счисления эта часть маски будет состоять из всех нулей. Значит, при поразрядном умножении на неё получится обязательно число из всех нулей. То есть, 0. Получаем, что последняя часть IP-адреса также очевидна: 0.

Остаётся рассмотреть третий слева третий байт. Переведём в двоичную систему третьи части IP-адреса и маски: $167 = 10100111$, $240 = 11110000$.

Запишем получившиеся двоичные числа друг под другом и поразрядно перемножим:

```

10100111
x 11110000
-----
10100000

```

Полученное двоичное число переведем в десятичную систему счисления: $10100000_2 = 160$.

Значит, адрес сети: 137.219.160.0. Найдём эти числа в таблице и запишем их в качестве ответа: CFDA.

Ответ: CFDA.

12.3.1. 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: _____.

12.3.2. 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: _____.

12.3.3. 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: 153.236.189.51

Маска: 255.255.224.0

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	51	153	160	189	224	236	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: _____.

12.3.4. 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: 198.213.231.73

Маска: 255.255.252.0

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	73	198	213	228	231	252	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: _____.

12.3.5. 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: 251.137.219.94

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	94	137	216	219	248	251	255

Пример.

Пусть искомый IP-адрес 192.168.128.0, и дана таблица

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>
128	168	255	8	127	0	17	192

В этом случае правильный ответ будет записан в виде: HBAF

Ответ: _____.

12.4. 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 207.176.101.81 адрес сети равен 207.176.96.0. Чему равно наибольшее значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Решение. Используем ту же идею, что была подробно рассмотрена в решении задания 12.3. Но, в отличие от задания 12.3, нам неизвестна маска сети, известен адрес сети, а маску нужно восстановить. Рассмотрим числа, соответствующие третьему слева байту IP-адресу и адресу сети (101 и 96). Второе число было получено применением поразрядной конъюнкции к двоичным разрядам первого числа и маски сети. Переведём оба числа в двоичное представление: $101 = 1100101_2$, $96 = 1100000_2$. Запишем каждое из этих чисел в 8-разрядном виде (числа получились семиразрядные, поэтому добавим перед каждым из них цифру 0). Затем запишем их друг под другом, как будто над первым числом произвели поразрядную конъюнкцию, и получили второе число:

```
01100101
0
abcdefgh
01100000
```

В качестве разрядов маски мы пока записали буквы от а до h. Это те разряды маски, которые нам нужно определить. Каждый из них может принимать значение 0 или 1.

Анализируем записанную операцию поразрядной конъюнкции. В разрядах *b* и *c* замечаем, что исходные единицы остались единицами. Умножение единицы на двоичный разряд может дать единицу только если этот двоичный разряд — единица. Значит, $b = 1$, $c = 1$. Также замечаем, что в разрядах *f* и *h* были единицы, а после конъюнкции стали нули. Значит, умножались на нули. То есть, $f = 0$, $h = 0$. На данный момент понимаем, что искомая маска: `a11de0g0`. Теперь вспомним, что по условию «в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули». Значит, если в маске где-то стоит 1, то все разряды левее 1 тоже должны быть обязательно единицами. А если в маске где-то стоит 0, то все разряды правее 0 тоже обязательно должны быть 0. Из этих соображений делаем вывод, что $a = 1$, $g = 0$. На данный момент искомая маска: `111de000`. Разряды *d* и *e* нам неизвестны, но по условию требуется найти наибольшее значение. Наибольшее значение будет в том случае, если эти разряды — единицы. Получаем маску: `11111000`. Переведем это число в десятичную систему счисления: $11111000_2 = 248$.

Ответ: 248.

❗ Если при решении забыть записать обрабатываемые числа в виде 8-разрядных двоичных чисел (в данном случае — не записать ведущие нули), то, вероятнее всего, в искомой маске будет потеряна ведущая единица, и будет получен неверный ответ.

12.4.1. 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 153.214.91.37 адрес сети равен 153.214.80.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Ответ: _____.

12.4.2. 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 153.214.115.37 адрес сети равен 153.214.96.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Ответ: _____.

12.4.3. 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 83.242.102.102 адрес сети равен 83.242.96.0. Чему равно наименьшее возможное значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Ответ: _____.

12.4.4. 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: _____.

12.4.5. 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: _____.

12.5. 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 231.117.213.243 адрес сети равен 231.117.208.0. Чему равно наименьшее возможное количество единиц в разрядах маски?

Решение. Заметим, что при преобразовании IP-адреса в адрес сети первые два байта остались неизменными, а последний байт превратился в 0. Значит, в маске сети первые два байта состояли из сплошных единиц (числа 255), а последний байт был равен нулю. Рассмотрим третий слева байт. Именно в нем число уменьшилось, но при этом не превратилось в 0.

Значит, часть разрядов третьего слева байта маски состоит из единиц, а часть — нули.

Переведём оба числа 3-го байта 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, $a = 1$, $b = 1$, $d = 1$. А разряды *f* и *h* были единицами, а стали нулями. Значит, $f = 0$, $h = 0$. Получаем текущий третий 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 наименьшим количеством единиц, оставшийся неизвестный разряд маски принимаем равным нулю. Получаем, что третий байт маски: 11110000 . В нём 4 единицы. Теперь нужно обязательно заметить, что в вопросе нас просят указать количество единиц в разрядах **маски**, а не количество единиц в третьем байте маски. А маска состоит из четырёх байт. Первые два байта, как мы уже поняли, состоят из сплошных единиц (по 8 единиц в байте), а последний байт — из нулей. Значит, всего единиц в маске: $8 + 8 + 4$.
Ответ: 20.

12.5.1. 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 183.142.167.102 адрес сети равен 183.142.160.0. Чему равно наименьшее возможное количество единиц в разрядах маски?

Ответ: _____.

12.5.2. 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 83.242.147.102 адрес сети равен 83.242.144.0. Чему равно наибольшее возможное количество единиц в разрядах маски?

Ответ: _____.

12.5.3. 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: _____.

12.5.4. 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 раз-

ряда — 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: _____.

12.5.5. 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 183.242.107.102 адрес сети равен 183.242.104.0. Чему равно наименьшее возможное количество единиц в разрядах маски?

Ответ: _____.

13.1. В соревнованиях по велоспорту участвует 200 спортсменов. Каждому из них выделяют специальный номер, идентифицирующий его в автоматической системе учета результата заезда. При регистрации участника для записи его номера система использует минимально возможное количество бит, одинаковое для каждого участника. Каков объем информации в битах, записанный устройством после регистрации 100 участников?

Решение. Для нахождения количества информации используем формулу $2^i \geq N$. Где N — количество равновероятных событий, i — наименьшее целое число, количество информации (бит) в сообщении об одном таком событии. Определим, какое из чисел (200 или 100) в данной задаче соответствует числу N . Заметим, что каждому спортсмену выделяют персональный номер, и именно эти номера затем кодируют. Значит, $N = 200$. Из неравенства $2^i \geq 200$ находим наименьшее i . Для этого находим наименьшую степень числа 2, которая больше или равна 200. Это 256. 256 — это 2 в восьмой степени. Значит, $i = 8$. Требуется определить объем информации при регистрации 100 участников. Умножаем 100 на 8.

Ответ: 800.

13.3.1. В некоторой стране автомобильный номер состоит из 7 символов: сначала 2 буквы, затем 3 цифры, затем еще 2 буквы. При этом буквы могут быть выбраны только из 12 строчных букв местного алфавита. Среди цифр не используются цифры 6 и 9. Автоматизированная система хранит номера автомобилей следующим образом. Используется посимвольное кодирование и в памяти системы для кодирования каждого символа используется минимально возможное и одинаково целое количество бит (для букв и цифр отдельно). А для хранения всего пароля используется минимально возможное целое количество байт. Какое количество информации (в байтах) требуется для хранения номеров 160 автомобилей?

Ответ: _____.

13.3.2. В некоторой стране автомобильный номер состоит из 9 символов: сначала 3 буквы, затем 4 цифры, затем еще 2 буквы. При этом буквы могут быть выбраны только из 25 строчных букв местного алфавита. Среди цифр не используются цифры 0 и 8. Автоматизированная система хранит номера автомобилей следующим образом. Используется посимвольное кодирование и в памяти системы для кодирования каждого символа используется минимально возможное и одинаково целое количество бит (для букв и цифр отдельно). А для хранения всего пароля используется минимально возможное целое количество байт. Какое количество информации (в байтах) требуется для хранения номеров 120 автомобилей?

Ответ: _____.

13.1.3. Для передачи секретного сообщения используется код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем сообщения (в байтах) длиной в 200 символов.

Ответ: _____.

13.1.4. Для передачи секретного сообщения используется код, состоящий из латинских строчных букв. При этом все буквы кодируются одним и тем же (минимально возможным) количеством бит. Определите информационный объем сообщения длиной в 100 символов (в битах).

Ответ: _____.

13.1.5. При передаче сообщения используются только десятичные цифры и строчные буквы местного алфавита. В местном алфавите 15 букв. При этом используется посимвольное кодирование, и каждый символ кодируется минимально возможным количеством бит. Каков объем информации (в байтах), записанный устройством после передачи 200 символов?

Ответ: _____.

13.2. Для регистрации на сайте онлайн-магазина пользователю предлагается придумать пароль, состоящий из 5 символов. Правилами безопасности сайта разрешается использовать только строчные буквы латинского алфавита. 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?

Решение.

Анализируя задачу, определяем, что сначала кодируется каждый символ минимально возможным целым количеством бит. По формуле $2^i \geq N$ определим количество бит, которым кодируется каждый символ. Количество вариантов каждого символа — 26 (число букв латинского алфавита). Решаем неравенство: $2^i \geq 26$. Получаем наименьшее $i = 5$ бит на каждый символ. Всего в пароле 5 символов. Значит, для записи пароля требуется $5 \cdot 5 = 25$ бит. По условию пароль кодируется минимально возможным количеством байт. Переведем 25 бит в байты. Поделим для этого 25 на 8. Получаем 3 целых и $1/8$. Минимальное количество байт, в которое это поместится — 4. Умножаем число пользователей (40) на количество требуемых байт (4).

Ответ: 160.

13.2.1. 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: _____.

13.2.2. 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: _____.

13.2.3. 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: _____.

13.2.4. 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: _____.

13.2.5. При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 15 символов и содержащий только символы из 12-символьного набора: А, В, С, D, Е, F, G, Н, К, L, М, 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: _____.

Общие замечание в решении задач 13.3.

Задания решаются аналогично заданию 13.2. В некоторых заданиях нужно обратить внимание на то, что, в отличие от задач 13.2, отдельные элементы кодируются по отдельности (например, буквы и цифры, номера кресел и номера рядов). Это значит, что при вычислении требуемого количества информации нужно не объединять в единое количество N число вариантов того и другого, а отдельно применять для этих элементов формулу $2^i \geq N$.

13.3.1. В некоторой стране автомобильный номер состоит из 7 символов: сначала 2 буквы, затем 3 цифры, затем ещё 2 буквы. При этом буквы могут быть выбраны только из 12 строчных

букв местного алфавита. Среди цифр не используются цифры 6 и 9. Автоматизированная система хранит номера автомобилей следующим образом. Используется посимвольное кодирование, и в памяти системы для кодирования каждого символа используется минимально возможное и одинаково целое количество бит (для букв и цифр отдельно). А для хранения всего пароля используется минимально возможное целое количество байт. Какое количество информации (в байтах) требуется для хранения номеров 160 автомобилей?

Ответ: _____.

13.3.2. В некоторой стране автомобильный номер состоит из 9 символов: сначала 3 буквы, затем 4 цифры, затем ещё 2 буквы. При этом буквы могут быть выбраны только из 25 строчных букв местного алфавита. Среди цифр не используются цифры 0 и 8. Автоматизированная система хранит номера автомобилей следующим образом. Используется посимвольное кодирование, и в памяти системы для кодирования каждого символа используется минимально возможное и одинаково целое количество бит (для букв и цифр отдельно). А для хранения всего пароля используется минимально возможное целое количество байт. Какое количество информации (в байтах) требуется для хранения номеров 120 автомобилей?

Ответ: _____.

13.3.3. 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: _____.

13.3.4. На стадионе имеется 10 секторов, в каждом из которых установлено по 80 кресел в 40 рядов. При кодировании но-

мера 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: _____.

13.3.5. Кабельная сеть проводит голосование среди зрителей о том, какой из четырёх 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

ИСПОЛНЕНИЕ ЦИКЛИЧЕСКОГО АЛГОРИТМА ДЛЯ ФОРМАЛЬНОГО ИСПОЛНИТЕЛЯ

Общее условие для задач 14.1. 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).

14.1. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 73 идущих подряд цифр 4? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (444) ИЛИ нашлось (222)

ЕСЛИ нашлось (444)

ТО заменить (444, 2)

ИНАЧЕ заменить (222, 4)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Решение.

Этап 1. Анализируем программу. Понимаем, что до тех пор, пока в строке имеется комбинация «444», в первую очередь эта комбинация будет заменяться на «2». В исходной строке 73 цифры 4. Значит, в ней 24 полных троек «444». Каждая из них заменится на одну цифру 2. Получится строка из 24 подряд идущих цифр 2, и затем одной цифры 4. Но программа при этом не закончится (ведь в строке есть «444» или «222»).

Этап 2. Так как в строке теперь нет «444», то первые «222» заменятся на «4».

После этого в строке все также нет «444». Поэтому самые левые «222» (они стоят сразу после первой цифры 4) заменятся на «4».

После этого в строке все также нет «444». Поэтому самые левые «222» (они стоят сразу после второй 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, 2)». То есть, эти первые «444» превратятся в цифру 2.

Этап 3. Замечаем, что в результате Этапа 2 первые 9 цифр 2 заменились на одну цифру 2. То есть, из 9-ти цифр 2 в строке пропало 8 цифр 2. Будем по очереди «убирать» из строки 8 цифр 2, если в строке есть хотя бы 9 цифр 2.

Из 24-х цифр 2 останется 16. Из 16-ти цифр 2 останется 8.

Теперь нельзя сказать, что эти 8 цифр 2 пропадут, потому что это происходит только при хотя бы 9-ти цифрах 2. А их осталось только 8. Применим к этим цифрам программу без упрощений.

Самые левые «222» заменятся на «4». Получится «4222224» (Вы ведь не забыли во время всех этих размышлений с цифрами 2, что в их конце после Этапа 1 оставалась ещё одна цифра 4!). Теперь самые левые «222» заменятся на «4». Получится «44224». В этой строке нет ни одной комбинации «444» или «222», поэтому программа на этом закончится.

Ответ: 44224.

❗ Основная ошибка при решении этой задачи — увлечься обобщениями. Так, например, после Этапа 1 может показаться, что на Этапе 2 аналогично каждая тройка «222» сначала заменится на «4». Постарайтесь внимательнее отнестись к порядку проверки условия в задаче и связанных с этим действий! В данном случае было важно заметить, что условие «**нашлось** (444)» проверяется раньше, чем «**заменить** (222, 4)». Поэтому, как только в строке есть хотя бы одна «444», она сразу заменяется на «2», независимо от того, какое действие выполнялось только что, и сколько ещё в строке «222».

14.1.1. 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: _____.

14.1.2. 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: _____.

14.1.3. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 69 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: _____.

14.1.4. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 90 идущих подряд цифр 3 и в конце одной цифры 1? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (331) ИЛИ нашлось (166)

ЕСЛИ нашлось (331)

ТО заменить (331, 16)

ИНАЧЕ заменить (166, 31)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

14.1.5. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 90 идущих подряд цифр 2 и в конце одной цифры 7? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (227) ИЛИ нашлось (755)

ЕСЛИ нашлось (227)

ТО заменить (227, 75)

ИНАЧЕ заменить (755, 27)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

Замечание к решению задач 14.2. Задачи 14.2 решаются аналогично задаче 14.1. Основная идея правильного решения — разбить решение на этапы, внутри этапа обобщать действия. При этом внимательно следить за тем, какое условие (и связанное с ним действие) выполняются в первую очередь.

14.2.1. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 83 идущих подряд цифр 7? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (77777) ИЛИ нашлось (222)

ЕСЛИ нашлось (77777)

ТО заменить (77777, 22)

ИНАЧЕ заменить (222, 2)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

14.2.2. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 76 идущих подряд цифр 3? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (33333) ИЛИ нашлось (111)

 ЕСЛИ нашлось (33333)

 ТО заменить (3333, 11)

 ИНАЧЕ заменить (111, 1)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

14.2.3. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 65 идущих подряд цифр 4 и в конце одной цифры 1? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (4441) ИЛИ нашлось (155)

 ЕСЛИ нашлось (4441)

 ТО заменить (4441, 15)

 ИНАЧЕ заменить (155, 41)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

14.2.4. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из цифры 3 и затем ещё 73 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (3888) ИЛИ нашлось (773)

 ЕСЛИ нашлось (388)

 ТО заменить (388, 73)

 ИНАЧЕ заменить (773, 38)

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

14.2.5. Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 73 идущих подряд цифр 6? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (6666) ИЛИ нашлось (111)

ЕСЛИ нашлось (666)

ТО заменить (666, 1)

ИНАЧЕ заменить (111, 6)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.

Общее условие для задач 14.3. 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).

14.3. 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>

ЕСЛИ <снизу свободно>

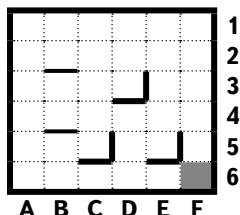
ТО вниз

ИНАЧЕ вправо

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

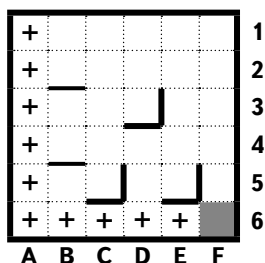
КОНЕЦ



Решение. Для того, чтобы не выполнять данную программу по отдельности для каждой из 36 клеток лабиринта, попытаемся составить обобщённые представления о том, как двигается РОБОТ по лабиринту. Заметим, что пока справа или снизу от РОБОТа есть свободная клетка, РОБОТ пытается продвинуться вниз. А если снизу не свободно, двигается вправо. Начнём с клетки A1. Снизу от неё свободно, РОБОТ передвигается в клетку A2. Команда **вправо** не выполняется, потому что она написана после ИНАЧЕ, а у нас выполненлся случай ТО. Снизу от A2 также свободно, поэтому РОБОТ передвигается вниз, в A3. Очевидно, так будет продолжаться до клетки A6. В ней условие **снизу свободно** уже перестанет выполняться, и РОБОТ передвинется **вправо**, в клетку B6. В ней аналогично снизу будет не свободно, и РОБОТ передвинется в клетку C6. Так РОБОТ «доберётся» до клетки F6, в которой перестанет выполняться условие цикла ПОКА (справа свободно ИЛИ снизу свободно), программа закончится и РОБОТ остановится в клетке F6. Делаем вывод, что клетка A1 удовлетворяет условию.

Анализируем, как вёл себя РОБОТ во всех пройденных клетках, и понимаем, что если бы он начинал движение в любой из них, он двигался бы далее точно так же. Значит, все клетки, через которые прошел РОБОТ, также будут удовлетворять условию.

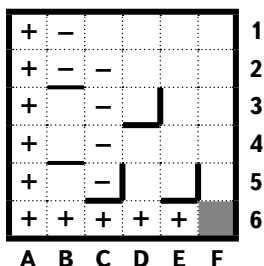
Отметим их в лабиринте «плюсиками», чтобы не забыть о них при подсчёте итогового значения.



Заметим, что данное обобщение («также отметим как искомые все клетки, через которые проехал РОБОТ») не всегда будет верно. Например, если РОБОТ на каждом шаге цикла выполнял бы не одно, а несколько действий, такое обобщение было бы, вероятно, неверным. Теперь рассмотрим, как будет двигаться РОБОТ из клетки В1. В ней снизу свободно, поэтому РОБОТ движется **вниз**, в клетку В2. В ней снизу не свободно, поэтому РОБОТ движется **вправо**, в клетку С2.

❗ Важно понимать, что РОБОТ движется вправо только на одну клетку, а не вправо до ближайшей стены!

В клетке С2 снова снизу свободно, поэтому РОБОТ движется вниз, в клетку С3, из неё аналогично в клетку С4, из неё в клетку С5. В ней перестаёт выполняться условие «справа свободно ИЛИ снизу свободно», цикл заканчивается, программа останавливается. РОБОТ не остановился в клетке F6, как требовалось, поэтому клетка В1 нас не устраивает. Так как при анализе движения из клетки А1 мы сделали вывод, что все клетки, которые проезжает данный РОБОТ, ведут в ту же итоговую клетку, делаем вывод, что все клетки В2, С2, С3, С4, С5 также нас не устраивают. Отметим это в лабиринте:



Теперь можно попробовать сформулировать обобщённый алгоритм движения РОБОТА: двигаемся вниз, пока свободно. Если упираемся в препятствие и справа при этом свободно, делаем «шаг вправо». Так повторяем, пока не попадём в «тупик» (справа и снизу не свободно). По этой логике понимаем, что из клетки В3 уткнёмся в С5. Пометим В3 и В4 минусом. Из клетки В5 доедем до F6. Пометим В5 плюсом. Из клетки С1 попадём в тупик С5. Из клеток D1, D2 и D3 попадём в тупик D3. Из клеток E1–E5 попадём в тупик E5. А из клеток D4, D5, F1–F5 благополучно доедем до F6.

Теперь осталось только понять, следует ли считать как правильную саму клетку F6. По условию не сказано, что её не следует учитывать в ответе. Значит, её тоже нужно проанализировать. Начиная программу в клетке F6 условие цикла (справа свободно ИЛИ снизу свободно) сразу не выполнится, и программа сразу завершится.

Робот после выполнения программы окажется в клетке F6. Это то, что требуется. Клетка F6 также удовлетворяет условию.

+	-	-	-	-	+	1
+	-	-	-	-	+	2
+	-	-	-	-	+	3
+	-	-	-	-	+	4
+	+	-	+	-	+	5
+	+	+	+	+	+	6
A	B	C	D	E	F	

Аккуратно посчитаем количество плюсигов.
Ответ: 18.

14.3.1. 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>
 ЕСЛИ <справа свободно>
 ТО вправо
 ИНАЧЕ вниз
 КОНЕЦ ЕСЛИ
 КОНЕЦ ПОКА
 КОНЕЦ

						1
						2
						3
						4
						5
						6
A	B	C	D	E	F	

Ответ: _____.

14.3.2. 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>
 ЕСЛИ <справа свободно> вправо
 вниз
 КОНЕЦ ПОКА
 КОНЕЦ

						1
						2
						3
						4
						5
						6
A	B	C	D	E	F	

Ответ: _____.

14.3.3. 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>

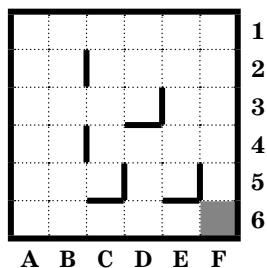
 вниз

 КОНЕЦ ПОКА

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.



14.3.4. 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>

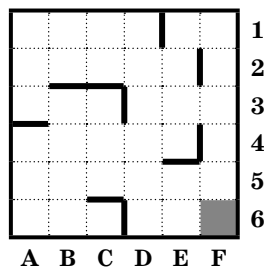
 вправо

 КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.



14.3.5. 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

ЕСЛИ <снизу свободно>

вниз

ИНАЧЕ

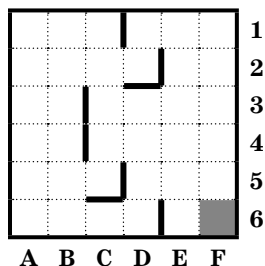
вверх

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Ответ: _____.



Общее условие для задач 14.4. 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).

14.4. Чертёжнику был дан для исполнения следующий алгоритм (буквами n, a, b обозначены неизвестные числа, при этом $n > 1$):

НАЧАЛО

сместиться на $(3, -9)$

ПОВТОРИ n РАЗ

сместиться на (a, b)

сместиться на $(12, 8)$

КОНЕЦ ПОВТОРИ

сместиться на $(30, -35)$

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.

Решение.

Анализируя систему команд Чертёжника понимаем, что перемещение по осям x и y никак не зависит друг от друга. То есть, смещение по ним можно анализировать по отдельности.

Выпишем в виде выражения общее смещение Чертёжника по каждой оси:

По оси x : $3 + n(a + 12) + 30$

По оси y : $-9 + n(b + 8) - 35$

Так как Чертёжник должен возвратиться в исходную точку, его смещение по каждой оси должно быть равно нулю. Приравняем каждое выражение к нулю. Получим систему двух уравнений:

$$3 + n(a + 12) + 30 = 0$$

$$-9 + n(b + 8) - 35 = 0$$

В ней три неизвестных. Однако, нам не нужно искать значения всех переменных (a , b и n). Нам нужно найти только n . И мы знаем, что это целое число. Перенесём свободные слагаемые в правую часть уравнений:

$$n(a + 12) = -33$$

$$n(b + 8) = 44$$

Видим, что число n является делителем числа -33 и числа 44 . Так нам требуется наименьшее значение n , которое больше 1, найдём наименьший делитель чисел -33 и 44 . Это число 11.

Ответ: 11.

14.4.1. Чертёжнику был дан для исполнения следующий алгоритм (буквами n , a , b обозначены неизвестные числа, при этом $n > 1$):

НАЧАЛО

 сместиться на (4, 2)

ПОВТОРИ n **РАЗ**

 сместиться на (a, b)

 сместиться на (12, 8)

КОНЕЦ ПОВТОРИ

 сместиться на (-25, -37)

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: _____.

14.4.2. 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 Чертёжник возвратится в исходную точку.

Ответ: _____.

14.4.3. 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: _____.

14.4.4. Чертёжнику был дан для исполнения следующий алгоритм (буквами n , a , b обозначены неизвестные числа, при этом $n > 1$):

НАЧАЛО

 сместиться на $(18, -7)$

ПОВТОРИ n **РАЗ**

 сместиться на (a, b)

 сместиться на $(14, 7)$

КОНЕЦ ПОВТОРИ

 сместиться на $(-58, -18)$

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: _____.

14.4.5. Чертёжнику был дан для исполнения следующий алгоритм (буквами n , a , b обозначены неизвестные числа, при этом $n > 1$):

НАЧАЛО

 сместиться на $(9, -10)$

ПОВТОРИ n **РАЗ**

 сместиться на (a, b)

 сместиться на $(11, 9)$

КОНЕЦ ПОВТОРИ

 сместиться на $(-58, -18)$

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: _____.

ИНФОРМАЦИОННЫЕ МОДЕЛИ. ГРАФЫ. ВЫЧИСЛЕНИЕ КОЛИЧЕСТВА ПУТЕЙ НА ГРАФЕ

15.1. На карту нанесены 4 города (А, В и С).

Известно, что:

между городами А и С — три дороги,

между городами А и В — четыре дороги,

между городами В и С — две дороги.

По каждой из этих дорог можно ехать в обе стороны. Сколькими различными способами можно проехать из А в С, посещая каждый город не более одного раза?

Решение.

Из города А в город С можно попасть либо напрямую, либо через город В. Попасть напрямую — 3 способа (3 дороги из А в С).

Попасть через город В — 8 способов (из А в В — 4 дороги, из В в С — 2 дороги. Нужно проехать из А в В **и** из В в С. Варианты перемножаются. $4 \cdot 2 = 8$).

Оба рассмотренных варианта (напрямую **или** через город В) соединяются союзом **или**. Значит, эти варианты нужно сложить. То есть, $3 + 8 = 11$.

Ответ: 11.

15.1.1. На карту нанесены 4 города (А, В и С).

Известно, что:

между городами А и С — две дороги,

между городами А и В — три дороги,

между городами В и С — четыре дороги.

По каждой из этих дорог можно ехать в обе стороны. Сколькими различными способами можно проехать из А в С, посещая каждый город не более одного раза?

Ответ: _____.

15.1.2. На карту нанесены 4 города (А, В, С и D).

Известно, что:

между городами А и С — три дороги,

между городами А и В — две дороги,

между городами С и D — три дороги,
между городами В и D — четыре дороги.

По каждой из этих дорог можно ехать в обе стороны. Сколькими различными способами можно проехать из А в D, посещая каждый город не более одного раза?

Ответ: _____.

15.1.3. На карту нанесены 4 города (А, В, С и D).

Известно, что:

между городами А и С — три дороги,
между городами А и В — две дороги,
между городами А и D — две дороги,
между городами С и D — четыре дороги,
между городами В и D — три дороги.

По каждой из этих дорог можно ехать в обе стороны. Сколькими различными способами можно проехать из А в D, посещая каждый город не более одного раза?

Ответ: _____.

15.1.4. На карту, нанесены 4 города (А, В, С и D).

Известно, что:

между городами А и С — две дороги,
между городами А и В — три дороги,
между городами В и С — четыре дороги,
между городами С и D — три дороги,
между городами В и D — три дороги.

По каждой из этих дорог можно ехать в обе стороны. Сколькими различными способами можно проехать из А в D, посещая каждый город не более одного раза?

Ответ: _____.

15.1.5. На карту нанесены 4 города (А, В, С и D).

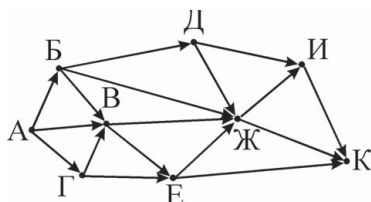
Известно, что:

между городами А и С — три дороги,
между городами С и В — две дороги,
между городами А и В — две дороги,
между городами С и D — две дороги,
между городами В и D — четыре дороги.

По каждой из этих дорог можно ехать в обе стороны. Сколькими различными способами можно проехать из А в D, посещая каждый город не более одного раза?

Ответ: _____.

15.2. 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?

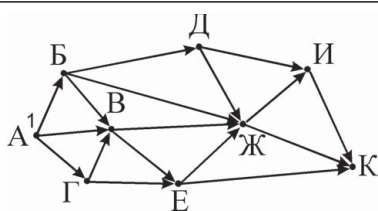


Решение. Традиционный на первый взгляд метод решения этой задачи — водить пальцем по схеме от города к городу и подсчитывать, сколькими различными способами это можно сделать. Этот способ по надёжности не выдерживает никакой критики — пропустить один или несколько вариантов очень легко, а проверить себя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

Используем 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 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).

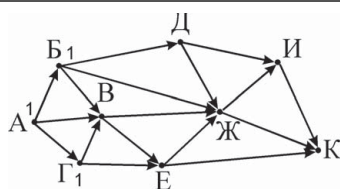
Будем последовательно искать, сколькими различными способами можно добраться из города **А** до каждого из городов, нарисованных на схеме.

Прежде всего, для выполнения нужного нам решения необходимо понять, сколькими способами

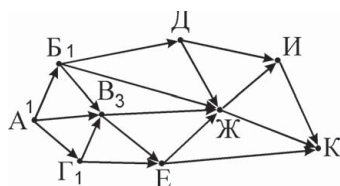


можно добраться из города **А** до самого города **А**. Важно понимать, что это можно сделать одним способом. Не нулём, как могло бы показаться поначалу. А именно одним способом — никуда не езда. Пометим вершину **А** числом 1 (количество способов, которым можно добраться до данной вершины).

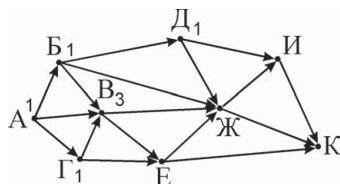
Теперь постараемся пометить какую-нибудь ещё вершину. Это можно сделать только для тех вершин, про которые известно нужное количество для всех вершин, входящих в данную вершину. Мы на данном этапе знаем нужное количество только для одной вершины — **А**. Значит, будем искать вершины, в которые входит 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**.



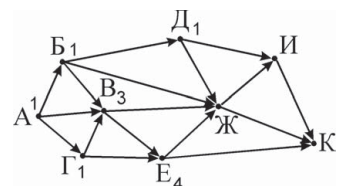
Опять будем искать город, про который для всех стрелок, входящих в него, на схеме уже написаны числа. Один из таких городов — город **В**. В него входит три стрелки. Для каждой из стрелок (из городов **А**, **Б**, **Г**) уже написаны числа. Нужное число для текущего города **В** — это сумма чисел на начальных сторонах стрелок. То есть, в город **В** можно прийти только по трём дорогам-стрелкам из города **А**, **Б** или **Г**. В каждый из них можно прийти одним способом. Значит, в город **В** можно прийти тремя способами. Пометим город **В** числом 3.



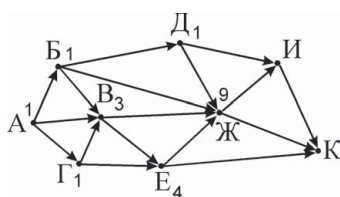
Найдём следующий город, для которого известны числа на началах всех входящих в него стрелках. Один из таких городов — **Д**. В него входит только одна стрелка из города **Б**. Значит, в город **Д** можно добраться столькими же способами — одним. Пометим город **Д** числом 1.



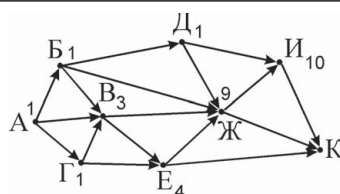
Найдём следующий город, про который известны числа из всех входящих стрелок. На данном этапе это только один город — **Е**. В него входит две стрелки-дороги (**В** и **Г**). Из города **В** — 3 способа, из города **Г** — 1 способ. Всего 4 способа. Пометим город **Е** числом 4.



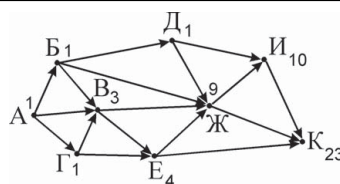
Ищем следующий город. На данном этапе это город **Ж**. В него входят 4 дороги. Про все эти стрелки-дороги известны числа для городов, из которых исходят стрелки. Это города **Д**, **Б**, **В**, **Е**. Рядом с ними написаны числа 1, 1, 3, 4. Сумма чисел — 9. Это количество способов (9) подписываем рядом 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**.

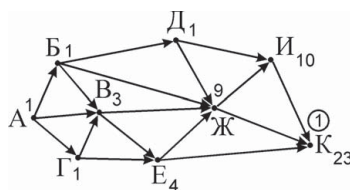


Последний город, для которого осталось посчитать нужное количество способов — город **К**. В него входит три стрелки-дороги (из городов **И**, **Ж**, **Е**). Рядом с ними написаны числа 10, 9, 4. Их сумма — 23. Это количество пишем рядом с городом **К**. Это и есть ответ.



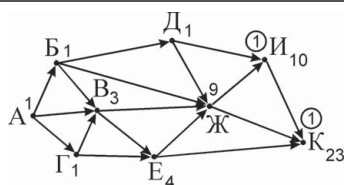
Приведённый алгоритм простой и эффективный. Однако при его выполнении можно легко сделать ошибку в арифметических вычислениях или случайно пропустить какую-нибудь стрелку, входящую в вершину. Необходимо сделать проверку. Выполним этот же алгоритм в обратную сторону. То есть, теперь будем искать количество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

Прежде всего обозначим количество способов, которыми 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.

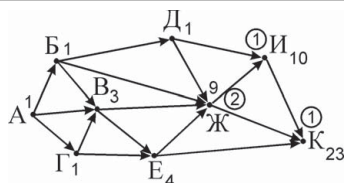


В прямом алгоритме мы искали все города, для которых были известны числа на началах всех входящих стрелок. Теперь мы будем искать города, для которых известны числа для всех исходящих стрелок.

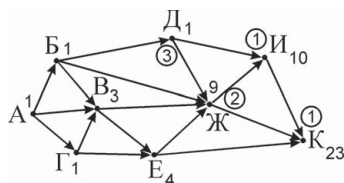
На данном шаге нас устроят только города, из которых выходит только одна стрелка — в город **К**. Такой город один — город **И**. Подпишем рядом с ним число 1 в кружке.



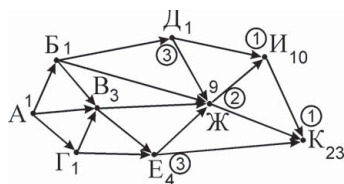
Теперь будем искать города, из которых все исходящие стрелки ведут в города с кружочками. Такая вершина одна — **Ж**. Из неё идут две стрелки — в города **И** и **К**. Рядом с ними в кружках написаны числа 1 и 1. Их сумма — 2. Подпишем рядом с вершиной **Ж** число 2 в кружке.



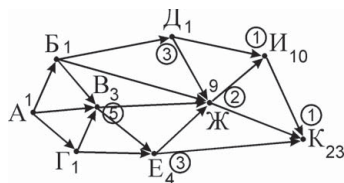
Ищем следующий город. Один из городов, у которого каждая исходящая стрелка приходит в город с кружком, — город **Д**. Из него выходят две стрелки (в города **Ж** и **И**). Числа на концах стрелок — 2 и 1. Их сумма — 3. Пишем число 3 в кружке рядом с городом **Д**.



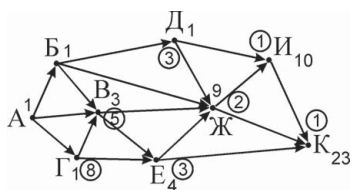
Следующий город — город **Е**. Из него выходит две стрелки (в города **Ж** и **К**). Числа в кружках на концах стрелок — 2 и 1. Их сумма — 3. Пишем число 3 в кружке рядом с городом **Е**.



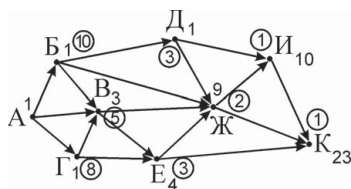
Теперь можно посчитать число вариантов для города **В**. Из него выходят две стрелки (в города **Е** и **Ж**). Числа в кружках на концах стрелок — 3 и 2. Их сумма — 5. Пишем число 5 в кружке возле города **В**.



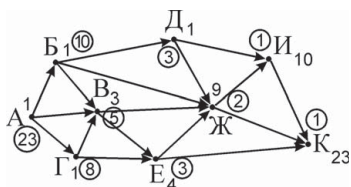
Теперь можно посчитать число вариантов для городов **Б** и **Г**. Начнём, например, с города **Г**. Из него выходят две стрелки (в города **В** и **Е**). Числа в кружках на концах стрелок — 5 и 3. Их сумма — 8. Пишем число 8 в кружке возле города **Г**.



Теперь посчитаем число вариантов для города **Б**. Из него выходят 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**.



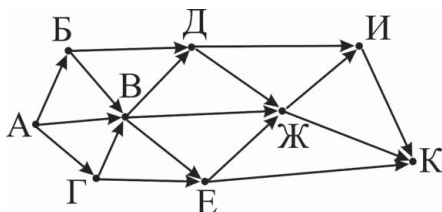
Наконец, можно посчитать число вариантов для города **А**. Из него выходят три стрелки-дороги (в города **Б**, **В**, **Г**). Числа в кружках на концах стрелок — 10, 5, 8. Их сумма — 23. Пишем это число в кружке возле города **А**. Это ответ. Результаты прямого и обратного алгоритма совпали. Значит, 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.

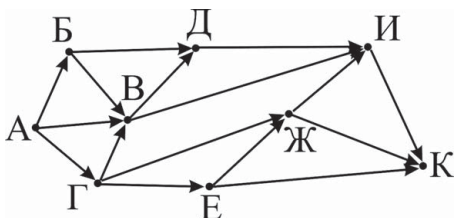
Эта рекомендация дана на основании наиболее частой ошибки — забыть добавить в сумму какое-нибудь из чисел или не заметить одну из стрелок. Поэтому, как правило, меньший результат — неверный. Это не значит, что нужно сразу принять за верный ответ большее из двух результатов. Это просто рекомендация начать поиск ошибки с меньшего результата. Для принятия решения о том, что найденный ответ — верный, результаты двух алгоритмов должны совпасть.

15.2.1. 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?



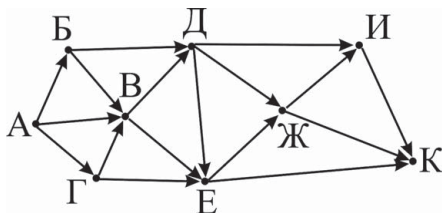
Ответ: _____.

15.2.2. 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?



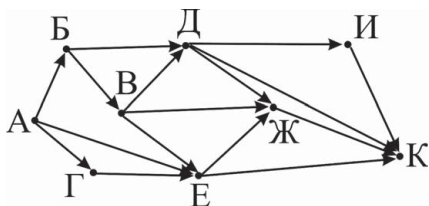
Ответ: _____.

15.2.3. 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?



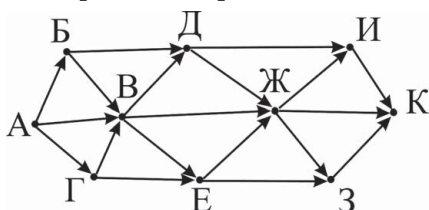
Ответ: _____.

15.2.4. 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: _____.

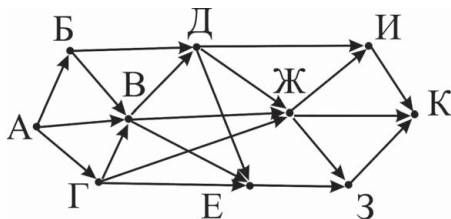
15.2.5. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Ответ: _____.

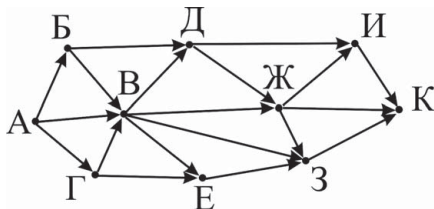
Замечание к решению задач 15.3. В этих задачах имеется дополнительное ограничение (проехать через определённый город или, наоборот, не проезжать через определённый город). Рекомендуем сначала вычеркнуть на схеме дороги, которые не удовлетворяют условию («объезжают» этот город, если через него нужно проехать или, наоборот, ведут в него/из него, если через него не нужно проезжать). Затем решить задачу аналогично решению задачи 15.2.

15.3.1. 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, не проходящих через город Е?



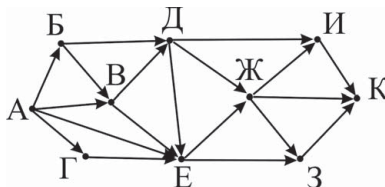
Ответ: _____.

15.3.2. 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, не проходящих через город З?



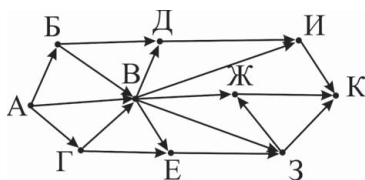
Ответ: _____.

15.3.3. 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?



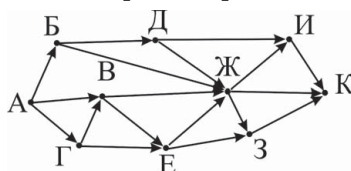
Ответ: _____.

15.3.4. 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 В?



Ответ: _____.

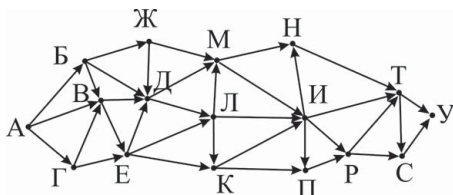
15.3.5. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К, проходящих через город Ж?



Ответ: _____.

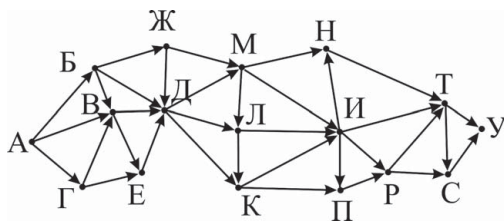
Замечание к решению задач 15.4. Так как требуется найти только пути, проходящие через определённый город (Л), рекомендуем сначала вычеркнуть на схеме все дороги, которые «объезжают» город Л. После этого решить задачу аналогично решению задачи 15.2. Можно несколько упростить решение, разбив его на два отдельных. Отдельно найти число вариантов из начального города А до города Л (двумя способами — «вперёд» и «назад»). Затем отдельно найти число вариантов из города Л до города У (также двумя способами — «вперёд» и «назад»). Затем найденные варианты перемножить.

15.4.1. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л, М, Н, П, Р, С, Т, У. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город У, проходящих через город Л?



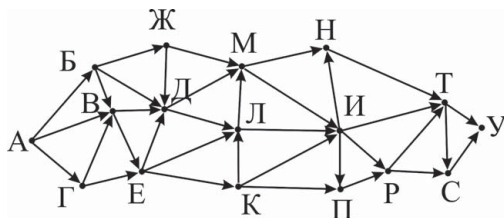
Ответ: _____.

15.4.2. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л, М, Н, П, Р, С, Т, У. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город У, проходящих через город Л?



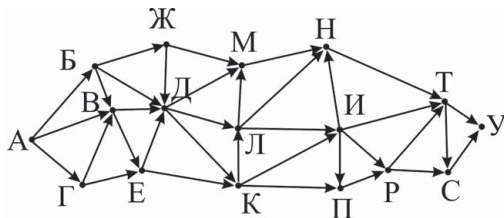
Ответ: _____.

15.4.3. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л, М, Н, П, Р, С, Т, У. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город У, проходящих через город Л?



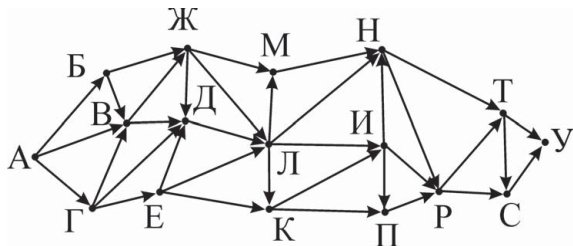
Ответ: _____.

15.4.4. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л, М, Н, П, Р, С, Т, У. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город У, проходящих через город Л?



Ответ: _____.

15.4.5. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л, М, Н, П, Р, С, Т, У. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город У, проходящих через город Л?



Ответ: _____.

16.1. В системе счисления с некоторым основанием десятичное число 50 записывается в виде 302. Укажите это основание.

Решение. Обозначим требуемое основание как N .

Переведём число 302_N в десятичную систему счисления. Получим: $302_N = 3 \cdot N^2 + 0 \cdot N^1 + 2 \cdot N^0 = 3N^2 + 2$. Приравняем это исходному числу 50. Решим полученное уравнение:

$3N^2 + 2 = 50 \Rightarrow 3N^2 = 48 \Rightarrow N^2 = 16 \Rightarrow N = 4$ (вариант $N = -4$ отбрасываем, потому что основание должно быть больше 1).

Ответ: 4.

16.1.1. В системе счисления с некоторым основанием десятичное число 51 записывается в виде 102. Укажите это основание.

Ответ: _____.

16.1.2. В системе счисления с некоторым основанием десятичное число 52 записывается в виде 202. Укажите это основание.

Ответ: _____.

16.1.3. В системе счисления с некоторым основанием десятичное число 123 записывается в виде 323. Укажите это основание.

Ответ: _____.

16.1.4. Запись десятичного числа в системах счисления с основаниями 4 и 7 в обоих случаях имеет последнюю цифру 0. Какое минимальное натуральное десятичное число удовлетворяет этому требованию?

Ответ: _____.

16.1.5. Запись десятичного числа в системах счисления с основаниями 6 и 15 в обоих случаях имеет последнюю цифру 0. Какое минимальное натуральное десятичное число удовлетворяет этому требованию?

Ответ: _____.

16.2. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 25 оканчивается на 4.

Решение.

Запись числа 25 оканчивается на 4 — это означает, что при делении числа 25 на основание системы счисления в остатке получается 4. $25 - 4 = 21$. То есть, основание системы счисления является делителем числа 21. Выпишем все делители числа 21: 1, 3, 7, 21. Так как 4 — остаток от деления на основание системы счисления, то основание системы счисления должно быть больше 4. Делители 1 и 3 нам не подходят, потому что они не больше числа 4. Остаются только 7 и 21.

Ответ: 7,21.

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

Ответ: _____.

16.2.2. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 29 оканчивается на 5.

Ответ: _____.

16.2.3. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 23 оканчивается на 5.

Ответ: _____.

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

Ответ: _____.

16.2.5. Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 25, запись которых в троичной системе счисления оканчивается на 21.

Ответ: _____.

16.3. Укажите наименьшее основание системы счисления, в которой запись десятичного числа 50 имеет ровно 3 значащих разряда.

Решение.

Обозначим основание системы счисления как N . Число 50 в этой системе счисления имеет 3 значащих разряда. То есть, оно трёхзначное. То есть, оно должно быть $\geq 100_N$ и $< 1000_N$. Переведём оба числа (100_N и 1000_N) в десятичную систему счисления. $100_N = N^2$, $1000_N = N^3$. Получаем: $N^2 \leq 50 < N^3$. Так нам нужно найти наименьшее целое N , можно просто перебрать возможные N , начиная с 2.

$N = 2$: $4 \leq 50 < 8$. Не подходит.

$N = 3$: $9 \leq 50 < 27$. Не подходит.

$N = 4$: $16 \leq 50 < 64$. Подходит.

Ответ: 4.

16.3.1. Укажите наименьшее основание системы счисления, в которой запись десятичного числа 70 имеет ровно 3 значащих разряда.

Ответ: _____.

16.3.2. Укажите наибольшее основание системы счисления, в которой запись десятичного числа 40 имеет ровно 3 значащих разряда.

Ответ: _____.

16.3.3. 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: _____.

16.3.4. 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: _____.

16.3.5. Запись числа 46_{10} в системе счисления с основанием N оканчивается на 1 и содержит 3 цифры. Чему равно основание этой системы счисления N ?

Ответ: _____.

16.4. Сколько единиц содержится в двоичной записи значения выражения: $4^{25} + 2^{61} - 31$?

Решение. Представим искомое выражение в виде суммы и разности степеней двойки. $4^{25} = 2^{50}$. $31 = 32 - 1 = 2^5 - 2^0$. Запишем полученные слагаемые в порядке убывания степеней:

$$2^{61} + 2^{50} - (2^5 - 2^0) = 2^{61} + 2^{50} - 2^5 + 2^0.$$

Исследуем, что за двоичное число получится, если сложить 2^x и 2^y ($x \neq y$). Рассмотрим это на примере $2^8 + 2^3$. $2^8 = 100000000_2$, $2^3 = 1000_2$. Найдём их сумму (столбиком):

$$\begin{array}{r} 1000000000 \\ + \quad \quad 1000 \\ \hline 100001000 \end{array}$$

Видим, что при сложении степеней двойки в двоичной системе получается число, состоящее из двух единиц, стоящих в соответствующих позициях.

Теперь исследуем, что за двоичное число получится, если вычесть 2^x и 2^y ($x > y$). Рассмотрим это на примере $2^8 - 2^3$. $2^8 = 100000000_2$, $2^3 = 1000_2$. Найдём их разность (столбиком):

$$\begin{array}{r} 1000000000 \\ - \quad \quad 1000 \\ \hline 11111000 \end{array}$$

Видим, что при вычислении $2^x - 2^y$ в двоичной системе получается число, состоящее из $(x - y)$ единиц, а затем из y нулей.

Рассмотрим наше выражение $2^{61} + 2^{50} - 2^5 + 2^0$. Видим в нём разность степеней $2^{50} - 2^5$, к которой слева и справа прибавляется степень числа 2. Количество единиц в двоичном представлении $2^{50} - 2^5$ равно $50 - 5 = 45$. К ним добавляется одна единица от числа 2^{61} и одна единица от числа 2^0 . Всего получаем $1 + 45 + 1 = 47$.

Ответ: 47.

16.4.1. Сколько единиц содержится в двоичной записи значения выражения: $8^{16} + 2^{52} - 15$?

Ответ: _____.

16.4.2. Сколько единиц содержится в двоичной записи значения выражения: $4^{40} - 8^{15} + 2^{54} + 7$?

Ответ: _____.

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

Ответ: _____.

16.4.4. Значение арифметического выражения: $25^{40} - 5^{30} + 24$ — записали в системе счисления с основанием 5. Сколько цифр «4» содержится в этой записи?

Ответ: _____.

16.4.5. Значение арифметического выражения: $16^{15} - 4^{13} + 2^{48} + 15$ — записали в системе счисления с основанием 4. Сколько цифр «3» содержится в этой записи?

Ответ: _____.

16.5. Какое восьмеричное число находится ровно посередине между числами 10010011_2 и $D7_{16}$? В ответе напишите только само число в 8-ричной системе счисления. Основание системы счисления писать не нужно.

Решение.

Число, находящееся ровно посередине между числами a и b , проще всего найти как $(a + b)/2$. Однако возникает вопрос, в какой системе счисления проще всего выполнить эти действия? С одной стороны, исходные числа, данные в двоичной и в 16-й системах счисления, проще всего сразу перевести в требуемую восьмеричную систему. Но операция деления на 2 в восьмеричной системе счисления весьма непроста. Вероятнее всего, для этого нужно будет знать восьмеричную таблицу умножения. Можно, конечно, перевести оба числа в 10-ю систему счисления, сделать в ней сложение и деление на 2, а затем перевести результат в 8-ю систему. Мы, однако, воспользуемся тем, что деление на 2 в двоичной системе счисления также весьма простое действие. Достаточно просто вычеркнуть из двоичного представления числа последнюю цифру.

Переведём число $D7_{16}$ в двоичную систему счисления. Для этого каждую его цифру запишем как 4-разрядное двоичное число: $1101\ 0111_2$. Сложим это число с исходным двоичным числом:

$$\begin{array}{r} +10010011 \\ \underline{11010111} \\ 101101010 \end{array}$$

Вычеркнем из получившегося числа последнюю цифру (10110101_2) и переведём полученное число в 8-ю систему счисления. Для этого сгруппируем разряды этого числа по 3 справа налево ($10\ 110\ 101_2$) и запишем вместо каждой группы разрядов соответствующую восьмеричную цифру: 265_8 .

Ответ: 265.

16.5.1. 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: _____.

16.5.2. 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: _____.

16.5.3. Какое шестнадцатеричное число находится ровно посередине между числами 57_8 и 1001001_2 ? В ответе напишите только само число в 16-ричной системе счисления. Основание системы счисления писать не нужно.

Ответ: _____.

16.5.4. Решите уравнение: $204_{N+1} = 204_N + 26_{16}$.

В ответе укажите значение переменной N.

Ответ: _____.

16.5.5. Решите уравнение: $305_{N+1} = 312_N + 2E_{16}$.

В ответе укажите значение переменной N.

Ответ: _____.

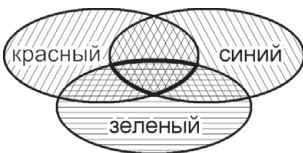
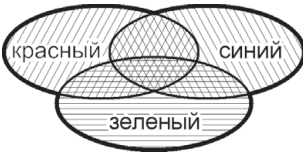
ПОИСК ИНФОРМАЦИИ В ИНТЕРНЕТЕ. АНАЛИЗ РЕЗУЛЬТАТА ПОИСКА ПО СЛОЖНОМУ УСЛОВИЮ

17.1. В таблице приведены запросы к поисковому серверу. Для обозначения логической операции «ИЛИ» в запросе используется символ |, а для логической операции «И» — &. Для каждого запроса указан его код — цифра от 1 до 4. Расположите номера запросов слева направо в порядке **возрастания** количества страниц, которые найдёт поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц.

Код	Запрос
1	красный & синий & зелёный
2	красный синий зелёный
3	красный & синий
4	синий зелёный

Решение.

Нарисуем диаграммы Эйлера-Венна для каждого запроса. Так как после этого нам нужно будет сравнить заштрихованные площади для всех четырёх запросов, будем рисовать в каждом запросе все три пересекающиеся области («красный», «зелёный» и «синий»), даже если какого-то слова в запросе нет.

	<p>1) красный & синий & зелёный Результатом запроса является область, в которой пересекаются все три штриховки</p>
	<p>2) красный синий зелёный Результатом запроса является область, в которой имеется хотя бы одна штриховка</p>

	<p>3) красный & синий Результатом запроса является область, в которой пересекаются две штриховки («красный» и «синий»)</p>
	<p>4) синий зелёный Результатом запроса является область, в которой имеется хотя бы одна штриховка («синий» и «зелёный»)</p>

Сравним площади найденных областей. Так как в задаче требуется расположить запросы в порядке **возрастания** количества найденных страниц, выпишем номера запросов, начиная с самой маленькой области.

Ответ: 1342.

❗ Найденные области могут быть «вложены» последовательно одна в другую. Если это не так, то, скорее всего, вы неправильно решили задачу (другой, маловероятный вариант — задача построена некорректно и у нее нет однозначного решения).

17.1.1. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке **возрастания** количества страниц, которые найдёт поисковый сервер по каждому запросу. В ответе перечислите цифры без пробелов и знаков препинания.

Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&».

№	Запрос
1	кролики лисицы
2	(зайцы & кролики) (лисицы & волки)
3	зайцы & кролики & лисицы & волки
4	зайцы & кролики

Ответ: _____.

17.1.2. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке **убывания** количества страниц, которые найдёт поисковый сервер по каждому запросу. В ответе перечислите цифры без пробелов и знаков препинания.

Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&».

№	Запрос
1	яблоки сливы
2	сливы (сливы & груши)
3	яблоки груши сливы
4	(яблоки груши) & сливы

Ответ: _____.

17.1.3. В таблице приведены запросы к поисковому серверу. Расположите номера запросов в порядке **убывания** количества страниц, которые найдёт поисковый сервер по каждому запросу.

Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» — «&».

№	Запрос
1	(яблоки груши) & сливы
2	яблоки сливы
3	сливы (сливы & груши)
4	яблоки груши сливы

Ответ: _____.

17.1.4. 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: _____.

17.1.5. 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: _____.

17.2. 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
зелёный & (красный синий)	25 600
синий & зелёный	7600
красный & синий & зелёный	2300

Какое количество страниц будет найдено по запросу:

красный & зелёный ?

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

Решение.

Для начала заметим, что каждый из рассматриваемых запросов содержит условие «& **зелёный**». Значит, на это выражение можно все запросы «сократить». То есть, будем решать задачу в виде:

Запрос	Количество найденных страниц
красный синий	25 600
синий	7600
красный & синий	2300

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

Для решения воспользуемся формулой включений и исключений для двух множеств: $K \mid C = K + C - K \& C$. Из этой формулы выразим требуемое неизвестное: $K = K \mid C - C + K \& C = 25\,600 - 7600 + 2300 = 20\,300$.

Ответ: 20 300.

17.2.1. 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: _____.

17.2.2. 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: _____.

17.2.3. 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
Колбаса & Молоко	15100
Сыр & Молоко	8900
Колбаса & Сыр & Молоко	7500

Какое количество страниц будет найдено по запросу:
(Колбаса | Сыр) & Молоко ?

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

Ответ: _____.

17.2.4. 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
Банан & Лимон	6700
Банан & Яблоко & Лимон	4300
(Банан & Яблоко) (Банан & Лимон)	15600

Какое количество страниц будет найдено по запросу:
Банан & Яблоко ?

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

Ответ: _____.

17.2.5. 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
Колёса & Оси	8700
Шины & Оси	4300
Колёса & Шины & Оси	1400

Какое количество страниц будет найдено по запросу:
(Колёса | Шины) & Оси ?

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

Ответ: _____.

Рекомендация к решению задач 17.3. Эти задания решаются так же, как задача 17.2, по формуле включений и исключений, только для трёх переменных:

$$A \mid B \mid C = A + B + C - A \& B - A \& C - B \& C + A \& B \& C$$

17.3.1. 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: _____.

17.3.2. 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)
Арбузы Дыни Тыквы	18
Дыни	7
Тыквы	9
Арбузы & Дыни & Тыквы	3
Арбузы & Дыни	5
Арбузы & Тыквы	6
Дыни Тыквы	12

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

Ответ: _____.

17.3.3. 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)
Стекло	26
Бетон	25
Кирпич	31
Стекло & Бетон	10
Стекло & Кирпич	16
Бетон & Кирпич	12
Стекло & Бетон & Кирпич	3

Какое количество страниц (в тысячах) будет найдено этой поисковой системой по запросу **Стекло | Бетон | Кирпич** ?

Ответ: _____.

17.3.4. 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)
Арбузы Дыни Тыквы	32
Дыни	9
Тыквы	11
Арбузы & Дыни & Тыквы	2
Арбузы & Дыни	3
Арбузы & Тыквы	7
Дыни Тыквы	15

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

Ответ: _____.

17.3.5. 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: _____.

18.1. В этом разделе собраны несколько разных задач, которые уже не встречаются в современном ЕГЭ. Однако их решение представляется полезным для подготовки к экзамену. Первая задача может быть решена просто подстановкой вариантов ответа в выражение и проверкой их правильности.

Следующие задачи рекомендуется решать через преобразование выражений по законам алгебры логики и избавления при этом от импликаций и отрицаний. Должны остаться только конъюнкция и дизъюнкция.

В задаче 3 при этом требуется ложность высказывания. Поэтому рекомендуется сделать отрицание всего выражения, чтобы свести его к проверке истинности.

После этого в задачах 2 и 3 нужно будет записать неравенства в виде системы и совокупности и решить алгебраически.

А в задачах 4 и 5 нужно будет осознать, чему равносильно выражение вида («число делится нацело на X» И «число делится нацело на Y»).

18.1.1. Какие из приведённых названий животных удовлетворяют логическому условию:

(первая буква гласная \rightarrow вторая буква гласная) \wedge

\wedge (последняя буква согласная \rightarrow предпоследняя буква согласная)

- 1) СТРАУС
- 2) АНТИЛОПА
- 3) ЖИРАФ
- 4) ДОДО
- 5) ТИГР

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

Ответ: _____.

18.1.2. Для какого целого числа X **ложно** высказывание:
 $((X > 1) \rightarrow (X > 3)) \vee (X = 3)$?

Ответ: _____.

18.1.3. Какое количество натуральных чисел удовлетворяет логическому условию:

$$\neg(X^2 > 9) \vee \neg((X < 7) \vee (X \geq 10)) ?$$

Ответ: _____.

18.1.4. Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наименьшего натурального числа A формула:
 $(\neg \text{ДЕЛ}(x, 12) \vee \neg \text{ДЕЛ}(x, 18)) \rightarrow \neg \text{ДЕЛ}(x, A)$

тождественно истинна (т. е. принимает значение 1 при любом натуральном значении переменной x)?

Ответ: _____.

18.1.5. Обозначим через $\text{ДЕЛ}(n, m)$ утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наибольшего натурального числа A формула:

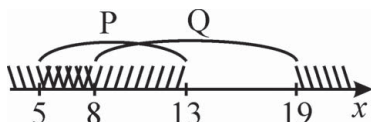
$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 8))$$

тождественно истинна (т. е. принимает значение 1 при любом натуральном значении переменной x)?

Ответ: _____.

18.2. На числовой прямой даны отрезки $P = [5, 13]$ и $Q = [8, 19]$. Укажите наименьшую возможную длину такого отрезка A , что формула $((x \in P) \rightarrow (x \in A)) \rightarrow ((x \in Q) \rightarrow (x \in A))$ верна при любых значениях x .

Решение. Приведённая формула достаточно сложна. Первым шагом решения упростим её. Для удобства этого действия введём новые обозначения. Обозначим выражения вида $(x \in M)$ просто как M (где M — A , P или Q). Получим: $(P \rightarrow A) \rightarrow (Q \rightarrow A)$. Избавимся от импликаций по формуле « $A \rightarrow B = \neg A \vee B$ »: $\neg(\neg P \vee A) \vee (\neg Q \vee A)$. Применим закон де Моргана (и закон отрицания отрицания): $P \wedge \neg A \vee \neg Q \vee A$. Переместительный закон: $P \wedge \neg A \vee A \vee \neg Q$. Закон поглощения: $P \vee A \vee \neg Q$. Снова переместительный: $A \vee (P \vee \neg Q)$. Нарисуем на числовой оси P , Q и $\neg Q$. Заштрихуем P и $\neg Q$.



Множества P и $\neg Q$ нужно объединить операцией \vee . Поэтому результатом $P \vee \neg Q$ будет являться вся часть числовой прямой, над которой есть хотя бы какая-нибудь штриховка. Получается, что под условие $P \vee \neg Q$ (то есть, $(x \in P) \vee \neg(x \in Q)$) не попадают только значения x , лежащие на полуотрезке $(13, 19]$. Нам требуется найти такой отрезок A , что добавление $(x \in A)$ к условию $(x \in P) \vee \neg(x \in Q)$ будет выполняться при всех значениях x . Очевидно, что если область $(13, 19]$ будет находиться внутри отрезка A , то требуемое условие выполнится. Отрезок A наименьшей длины, при котором область $(13, 19]$ будет находиться внутри отрезка A — это отрезок $[13, 19]$. Его длина $19 - 13 = 6$.

Ответ: 6.

18.2.1. На числовой прямой даны отрезки $P = [5, 13]$ и $Q = [8, 19]$. Укажите наименьшую возможную длину такого отрезка A , что формула $(\neg(x \in P) \rightarrow (x \in Q)) \rightarrow (x \in A)$ верна при любых значениях x .

Ответ: _____.

18.2.2. На числовой прямой даны отрезки $P = [5, 13]$ и $Q = [8, 19]$. Укажите наименьшую возможную длину такого отрезка A , что формула $((x \in Q) \rightarrow (x \in P)) \vee (x \in A)$ верна при любых значениях x .

Ответ: _____.

18.2.3. На числовой прямой даны отрезки $P = [5, 13]$ и $Q = [8, 19]$. Укажите наибольшую возможную длину такого отрезка A , что формула $\neg((x \in P) \rightarrow \neg(x \in Q)) \vee \neg(x \in A)$ верна при любых значениях x .

Ответ: _____.

18.2.4. На числовой прямой даны отрезки $P = [5, 13]$ и $Q = [8, 19]$. Укажите наибольшую возможную длину такого отрезка A , что формула $(x \in A) \rightarrow \neg(\neg(x \in P) \wedge \neg(x \in Q))$ верна при любых значениях x .

Ответ: _____.

18.2.5. На числовой прямой даны отрезки $P = [5, 13]$ и $Q = [8, 19]$. Укажите наименьшую возможную длину такого отрезка A , что формула $((x \in P) \wedge \neg(x \in A)) \rightarrow ((x \in Q) \wedge \neg(x \in A))$ верна при любых значениях x .

Ответ: _____.

18.3. 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:

$(x \& 106 \neq 0 \wedge x \& A = 0) \rightarrow x \& 96 \neq 0$

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

Решение. Упростим имеющееся выражение.

Часть 1. Избавимся от импликации:

$\neg(x \& 106 \neq 0 \wedge x \& A = 0) \vee x \& 26 \neq 0$.

Избавимся от отрицания перед скобкой (применим закон де Моргана):

$x \& 106 = 0 \vee x \& A \neq 0 \vee x \& 26 \neq 0$.

Часть 2. Исследуем выражение $x \& 106 = 0$. Переведем в двоичную систему счисления число 106 и запишем под полученным двоичным числом столбиком неизвестное нам двоичное число x . Его двоичные разряды обозначим буквами от a до g справа налево.

```

1101010
&
gfedcba
gf0d0b0

```

В результате поразрядной конъюнкции те разряды числа x , которые стоят под нулями числа 106, станут нулями, а те, которые стоят под единицами, останутся те же, что и были в числе x . Далее нас будут интересовать только они. Потому что по требованию $x \& 106 = 0$ все разряды результата должны быть нулями. Заметим, что если в числе x больше двоичных разрядов, чем в числе 106, то они тоже станут нулями. Поэтому они нас тоже не будут интересовать. Получаем, что условие $x \& 106 = 0$ равносильно условию, что разряды g, f, d и b числа x должны быть равны нулю. То есть, любое число x , в двоичном представлении которого разряды g, f, d и b равны нулю, удовлетворяет условию $x \& 106 = 0$.

Получается, для того чтобы общее условие задачи выполнялось, нужно чтобы оставшейся части выражения удовлетворяли все остальные числа x . То есть, такие, в двоичном представлении которых в разрядах g, f, d и b есть хотя бы одна единица.

Часть 3. Исследуем выражение $x \& 26 \neq 0$. Переведем в двоичную систему счисления число 26 и запишем под полученным двоичным числом столбиком неизвестное нам двоичное число x . Его двоичные разряды обозначим буквами от a до e справа налево.

```

11010
&
edcba
ed0b0

```

В результате поразрядной конъюнкции те разряды числа x , которые стоят под нулями числа 26, станут нулями, а те, которые стоят под единицами, останутся теми же, что и были в числе x . Далее нас будут интересовать только они, потому что по требованию $x \& 26 \neq 0$ хотя бы один разряд результата должен быть не нулем. Заметим, что если в числе x больше двоичных разрядов, чем в числе 26, то они станут нулями. Поэтому они нас тоже не будут интересовать. Получаем, что условие $x \& 26 \neq 0$ равносильно условию, что среди разрядов e, d и b числа x должна быть хотя бы одна единица. Но в части 1 мы выяснили, что нам осталось рассмотреть только числа x , у которых хотя бы одна единица есть в разрядах g, f, d и b . Получается, что из этих чисел условие $x \& 26 \neq 0$ «покрывает» числа единицами в разрядах b или d . То есть, остаются только такие числа x , в двоичном представлении которых в разрядах g или f есть хотя бы одна единица.

Часть 4. Условие $x \& A \neq 0$ выполняется для чисел x , в двоичном представлении которых есть хотя бы одна единица в одном из единичных разрядов числа A . Значит, наименьшим числом A , которое будет нас устраивать, будет двоичное число $gf00000$. Переведём его в двоичную систему счисления. Получим число $2^6 + 2^5 = 64 + 32 = 96$.

Ответ: 96.

❗ Все приведённые в решении сложные умозаключения нужны лишь для того, чтобы объяснить решение задачи. Для нахождения ответа в большинстве случаев этого не требуется. То есть, если логическое выражение в условии сводится к виду:

$x \& B = 0 \vee x \& A \neq 0 \vee x \& C \neq 0$, где B и C — некоторые числа, то для получения ответа нужно сделать следующее:

1. Разложить числа B и C на сумму степеней двойки.

В данной задаче: $B = 106 = 64 + 32 + 8 + 2$. $C = 26 = 16 + 8 + 2$.

2. Из множества чисел, составляющих сумму В вычесть множество чисел, составляющих сумму С.

Здесь под операцией «вычесть» подразумевается операция вычитания над множествами. То есть, из списка объектов первого множества вычеркиваются объекты, которые есть во втором множестве.

Получаем: $\{64, 32, 8, 2\} \setminus \{16, 8, 2\} = \{64, 32\}$.

3. Сумма элементов полученного множества даст ответ:

$$64 + 32 = 96.$$

Ответ: 96.

18.3.1. 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 А формула $(x \& 45 \neq 0 \wedge x \& A = 0) \rightarrow x \& 33 \neq 0$ тождественно истинна (т. е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

18.3.2. 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 А формула $(x \& A = 0 \wedge x \& 36 = 0) \rightarrow x \& 46 = 0$ тождественно истинна (т. е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

18.3.3. 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 А формула $x \& 51 \neq 0 \rightarrow (x \& A = 0 \rightarrow x \& 25 \neq 0)$ тождественно истинна (т. е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

18.3.4. 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 $(x \& 43 \neq 0 \wedge x \& A = 0) \rightarrow x \& 14 \neq 0$ тождественно истинна (т. е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

18.3.5. 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 $(x \& A = 0 \wedge x \& 58 \neq 0) \rightarrow x \& 34 \neq 0$ тождественно истинна (т. е. принимает значение 1 при любом неотрицательном целом значении переменной x)?

Ответ: _____.

18.4. Укажите наименьшее целое значение A , при котором выражение:

$(x > 20) \vee (y > 30) \vee (5x + 4y < A)$ истинно для любых целых значений x и y .

Решение.

Анализируя исходное выражение, видим, что оно истинно при выполнении хотя бы одного условия: $(x > 20)$ или $(y > 30)$. Значит, для значений x и y , которые не удовлетворяют этому условию, значение выражения зависит исключительно от условия $(5x + 4y < A)$.

Определим, при каких x и y условие $(x > 20) \vee (y > 30)$ ложно. Для этого найдём отрицание к этому выражению: $\neg((x > 20) \vee (y > 30))$. По закону де Моргана это равно: $\neg(x > 20) \wedge \neg(y > 30)$. Внесём отрицания в скобки и инвертируем операции сравнения. Получаем: $(x \leq 20) \wedge (y \leq 30)$. Учитывая эти ограничения, оценим выражение $5x + 4y$: $5x + 4y \leq 5 \cdot 20 + 4 \cdot 30 = 220$. Получаем, что $5x + 4y \leq 220$. Значит, мы ищем такое A , что $(220 \leq A)$. Отсюда получаем, что наименьшее целое A , при котором это условие гарантированно выполняется, равно 221.

Ответ: 221.

18.4.1. 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: _____.

18.4.2. 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: _____.

18.4.3. Для какого наименьшего целого числа A формула

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

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

Ответ: _____.

18.4.4. Для какого наибольшего целого числа A формула

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

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

Ответ: _____.

18.4.5. 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

ВЫПОЛНЕНИЕ АЛГОРИТМА, ОБРАБАТЫВАЮЩЕГО МАССИВ

19.1. 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	Алгоритмический язык
<pre>k=0 FOR i=0 TO 9 IF A(i) MOD 2=0 THEN k=k+1 END IF NEXT i</pre>	<pre>k:=0 нц для i от 0 до 9 если mod(A[i],2)=0 то k:=k+1 все кц</pre>
C++	Паскаль
<pre>k=0; for (i=0 ; i<=9 ; i++) if (A[i]%2==0) k=k+1;</pre>	<pre>k:=0; for i:=0 to 9 do if A[i] mod 2=0 then k:=k+1;</pre>
Python	
<pre>k=0 for i in range(10): if A[i]%2==0: k=k+1</pre>	

Решение.

Метод 1. Выполним трассировку программы «вручную», т.е., будем пошагово выполнять программу, отслеживая значения всех переменных.

Заметим, что в программе меняются только переменные k и i . Значения элементов массива не меняются. Для удобства запишем в таблицу значения элементов массива с их номерами, чтобы не ошибиться:

№	0	1	2	3	4	5	6	7	8	9
$A[№]$	4	7	6	4	2	3	5	10	7	8

Таблица трассировки:

Оператор	Условие	Переменные		Пояснения
		k	i	
k:=0;		0		Переменная k получает начальное значение, равное нулю.
for i:=0 to 9 do	0≤9? Да		0	Переменная i получает начальное значение счётчика цикла, равное 0.
if A[i] mod 2=0 then	0=0? Да			A[0] mod 2 = 4 mod 2 = 0
k:=k+1;		1		Переменная k увеличивается на 1
for i:=0 to 9 do	1≤9? Да		1	Переменная i (счётчик цикла) увеличивается на 1.
if A[i] mod 2=0 then	1=0? Нет			A[1] mod 2 = 7 mod 2 = 1
for i:=0 to 9 do	2≤9? Да		2	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	0=0? Да			A[2] mod 2 = 6 mod 2 = 0
k:=k+1;		2		Переменная k увеличивается на 1
for i:=0 to 9 do	3≤9? Да		3	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	0=0? Да			A[3] mod 2 = 4 mod 2 = 0
k:=k+1;		3		Переменная k увеличивается на 1
for i:=0 to 9 do	4≤9? Да		4	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	0=0? Да			A[4] mod 2 = 2 mod 2 = 0
k:=k+1;		4		Переменная k увеличивается на 1
for i:=0 to 9 do	5≤9? Да		5	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	1=0? Нет			A[5] mod 2 = 3 mod 2 = 1
for i:=0 to 9 do	6≤9? Да		6	Переменная i увеличивается на 1.

if A[i] mod 2=0 then	1=0? Нет			A[6] mod 2 = 5 mod 2 = 1
for i:=0 to 9 do	7≤9? Да		7	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	0=0? Да			A[7] mod 2 = 10 mod 2 = 0
k:=k+1;		5		Переменная k увеличивается на 1
for i:=0 to 9 do	8≤9? Да		8	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	1=0? Нет			A[8] mod 2 = 7 mod 2 = 1
for i:=0 to 9 do	9≤9? Да		9	Переменная i увеличивается на 1.
if A[i] mod 2=0 then	0=0? Да			A[9] mod 2 = 8 mod 2 = 0
k:=k+1;		6		Переменная k увеличивается на 1
for i:=0 to 9 do	10≤9? Нет		10	Переменная i увеличивается на 1.
<p>Цикл закончился. Текущее значение переменной <i>k</i> равно 6. <i>Ответ:</i> 6.</p> <p>❗ Если Вы хорошо понимаете, как работает программа, вовсе не обязательно выписывать так подробно все действия трассировки (в особенности, первые два столбца). Достаточно «водить пальцем» по программе, чтобы отслеживать, какая сейчас выполняется строка, и в столбцах для переменных <i>k</i> и <i>i</i> записывать текущие значения этих переменных.</p> <p><u>Метод 2.</u> Если опыт программирования у Вас большой, можно проанализировать программу, чтобы понять, что именно она делает. В данном случае программа подсчитывает количество чётных элементов массива. Значит, достаточно посчитать количество чётных чисел в списке.</p> <p>❗ Способ хорош только для достаточно простых программ. Для относительно сложных случаев уже вызывает ошибки, потому что при этом легко не заметить какой-нибудь особенности и принять желаемое за действительное. Рекомендуем решать задачу обоими методами и сравнивать на равенство полученные результаты.</p>				

19.1.1. 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	Алгоритмический язык
<pre>k = 0 FOR i = 0 TO 9 IF A(i) MOD 2 <> 0 THEN k = k + A(i) END IF NEXT i</pre>	<pre>k := 0 нц для i от 0 до 9 если mod(A[i], 2) <> 0 то k := k + A[i] все кц</pre>
C++	Паскаль
<pre>k = 0; for (i = 0 ; i <= 9 ; i++) if (A[i] % 2 != 0) k = k + A[i];</pre>	<pre>k := 0; for i := 0 to 9 do if A[i] mod 2 <> 0 then k := k + A[i];</pre>
Python	
<pre>k = 0 max = A[0] for i in range(10): if A[i] % 2 != 0: k = k + A[i]</pre>	

Ответ: _____.

19.1.2. 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	Алгоритмический язык
<pre>k = 0 m = A[1] FOR i = 1 TO 9 IF A(i) > m THEN k = k + 1 END IF NEXT i</pre>	<pre>k := 0 m := A[1] нц для i от 1 до 9 если A[i] > m то k := k + 1 все кц</pre>

С++	Паскаль
<pre> k = 0; m = A[1]; for (i = 1 ; i <= 9 ; i++) if(A[i] > m) k = k + 1; </pre>	<pre> k := 0; m := A[1]; for i:=1 to 9 do if A[i] > m then k := k + 1; </pre>
Python	
<pre> k = 0 m = A[1] for i in range(1,10): if A[i] > m: k = k + 1 </pre>	

Ответ: _____.

19.1.3. 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	Алгоритмический язык
<pre> k = 0 m = A(0) FOR i = 1 TO 9 IF A(i) < m THEN k = k + A(i) END IF NEXT i </pre>	<pre> k := 0 m := A(0) нц для i от 1 до 9 если A[i] < m то k := k + A[i] все кц </pre>
С++	Паскаль
<pre> k = 0; m = A(0); for (i = 1 ; i <= 9 ; i++) if(A[i] < m) k = k + A[i]; </pre>	<pre> k := 0; m := A[0]; for i := 1 to 9 do if A[i] < m then k := k + A[i]; </pre>
Python	
<pre> k = 0 m = A[0] for i in range(1,10): if A[i] < m: k = k + A[i] </pre>	

Ответ: _____.

19.1.4. 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	Алгоритмический язык
<pre> k = 0 FOR i = 0 TO 9 IF A(i) MOD 2 > 0 THEN k = i END IF NEXT i </pre>	<pre> к := 0 нц для i от 0 до 9 если mod(A[i], 2) > 0 то к := i все кц </pre>
C++	Паскаль
<pre> k = 0; for (i = 0 ; i <= 9 ; i++) if (A[i] % 2 > 0) k = i; </pre>	<pre> k := 0; for i := 0 to 9 do if A[i] mod 2 > 0 then k := i; </pre>
Python	
<pre> k = 0 for i in range(10): if A[i] % 2 > 0: k = i </pre>	

Ответ: _____.

19.1.5. 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	Алгоритмический язык
<pre> k=0 FOR i=0 TO 9 IF A(i) MOD 2=0 AND k=0 THEN k=A(i) END IF NEXT i </pre>	<pre> к:=0 нц для i от 0 до 9 если mod(A[i], 2)=0 и k=0 то к:=A[i] все кц </pre>

С++	Паскаль
<pre>k=0; for (i=0 ; i<=9 ; i++) if (A[i]%2==0 && k==0) k=A[i];</pre>	<pre>k:=0; for i:=0 to 9 do if (A[i] mod 2=0) and (k=0) then k := A[i];</pre>
Python	
<pre>k = 0 for i in range(10): if A[i] % 2 == 0 and k == 0: k = A[i]</pre>	

Ответ: _____.

Замечание к решению заданий 19.2. Здесь собраны задачи, которые точно так же решаются Методом 1 (из задачи 19.1). Но могут вызывать ошибки при решении Методом 2, потому что программы очень похожи на привычные алгоритмы поиска, но делают, зачастую не совсем то, что кажется на первый взгляд.

19.2.1. В программе используется целочисленный массив *A* с индексами от 0 до 9. Значения элементов массива равны 4; 5; 4; 7; 10; 8; 9; 11; 6; 3 соответственно, т.е. $A[0] = 4$; $A[1] = 5$ и т.д.

Определите значение переменной *max1* после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

Бейсик	Алгоритмический язык
<pre>k = 0 max1 = A(0) FOR i = 1 TO 9 IF A(i) >= A(k) THEN max1 = A(i) END IF NEXT i</pre>	<pre>k := 0 max1 := A[0] нц для i от 1 до 9 если A[i] >= A[k] то max1 := A[i] все кц</pre>
С++	Паскаль
<pre>k = 0; max1 = A[0]; for (i = 1 ; i <= 9 ; i++) if(A[i] >= A[k]) max1 = A[i];</pre>	<pre>k := 0; max1 := A[0]; for i:=1 to 9 do if A[i]>=A[k] then max1 := A[i];</pre>

Python
<pre> k = 0 max1 = A[0] for i in range(1,10): if A[i] >= A[k]: max1 = A[i] </pre>

Ответ: _____.

19.2.2. 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	Алгоритмический язык
<pre> k = 0 min1 = A(0) FOR i = 1 TO 9 IF A(i) <= min1 THEN k = i END IF NEXT i </pre>	<pre> k := 0 min1 := A[0] нц для i от 1 до 9 если A[i] <= min1 то k := i все кц </pre>
C++	Паскаль
<pre> k = 0; min1 = A[0]; for (i = 1 ; i <= 9 ; i++) if(A[i] <= min1) k = i; </pre>	<pre> k := 0; min1 := A[0]; for i:=1 to 9 do if A[i] <= min1 then k := i; </pre>
Python	
<pre> k = 0 min1 = A[0] for i in range(1,10): if A[i] <= min1: k = i </pre>	

Ответ: _____.

19.2.3. 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	Алгоритмический язык
<pre> k = 0 m = A(0) FOR i = 1 TO 9 IF A(i) > m THEN k = i END IF NEXT i </pre>	<pre> k := 0 m := A[0] нц для i от 1 до 9 если A[i] > m то k := i все кц </pre>
C++	Паскаль
<pre> k = 0; m = A[0]; for (i = 1 ; i <= 9 ; i++) if(A[i] > m) k = i; </pre>	<pre> k := 0; m := A[0]; for i := 1 to 9 do if A[i] > m then k := i; </pre>
Python	
<pre> k = 0 m = A[0] for i in range(1,10): if A[i] > m: k = i </pre>	

Ответ: _____.

19.2.4. 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	Алгоритмический язык
<pre> k = 0 m = 0 FOR i = 0 TO 9 IF A(i) > m THEN k = k + 1 m = A(i) END IF NEXT i </pre>	<pre> k := 0 m := 0 нц для i от 0 до 9 если A[i] > m то k := k + 1 m := A[i] все кц </pre>

C++	Паскаль
<pre> k = 0; m = 0; for (i=0 ; i<=9 ; i++) if(A[i] > m) { k = k + 1; m = A[i]; } </pre>	<pre> k := 0; m := 0; for i := 0 to 9 do if A[i] > m then begin k := k + 1; m := A[i] end; </pre>
Python	
<pre> k = 0 m = 0 for i in range(10): if A[i] > m: k = k + 1 m = A[i] </pre>	

Ответ: _____.

19.2.5. 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	Алгоритмический язык
<pre> k = 0 m = A(0) FOR i = 1 TO 9 IF A(i) > m THEN m = A(i) k = k + 1 END IF NEXT i </pre>	<pre> 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 </pre>
C++	Паскаль
<pre> k = 0; m = A[0]; for (i=1 ; i<=9 ; i++) if (A[i] > m) { m = A[i]; k = k + 1; } </pre>	<pre> k := 0; m := A[0]; for i := 1 to 9 do if A[i] > m then begin m := A[i]; k := k + 1; end; </pre>

Python

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

Ответ: _____.

Замечание к решению заданий 19.3 и 19.4.

Здесь собраны задачи, в которых, помимо изменения отдельных переменных (таких как, например, i и k), также изменяются и элементы массива. При решении этих задач Методом 1 (трассировкой, см. решение задачи 19.1), необходимо отслеживать в таблице трассировки не только значения отдельных переменных, но также и значения для элементов всего массива. Мы рекомендуем не игнорировать это действие, т. е. аккуратно нарисовать таблицу для всех используемых переменных и для всех десяти элементов массива, и аккуратно записывать в ней как начальные значения, так и все изменения. Эти задачи очень коварны. Часто из-за непрерывно меняющегося в течении цикла массива итоговое значение искомой переменной оказывается совсем не тем, чем кажущееся на первый взгляд.

Рекомендуем использовать при трассировке примерно такую таблицу:

i	c	k	Массив A									
			0	1	2	3	4	5	6	7	8	9
			4	7	6	4	2	3	5	10	7	8

19.3.1. 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 c после выполнения следующего фрагмента программы, записанной ниже на разных языках программирования.

Бейсик	Алгоритмический язык
<pre> c = 0 FOR i = 0 TO 4 IF A(i) < A(9-i) THEN c = c + 1 k = A(i) A(i) = A(9-i) A(9-i) = k END IF NEXT i </pre>	<pre> с := 0 нц для i от 0 до 4 если A[i] < A[9-i] то с := с + 1 k := A[i] A[i] := A[9-i] A[9-i] := k все кц </pre>
C++	Паскаль
<pre> c = 0; for (i = 0 ; i <= 4 ; i++) if (A[i] < A[9-i]) { c = c + 1; k = A[i]; A[i] = A[9-i]; A[9-i] = k; } </pre>	<pre> с := 0; for i := 0 to 4 do if A[i] < A[9-i] then begin с := с + 1; k := A[i]; A[i] := A[9-i]; A[9-i] := k end; </pre>
Python	
<pre> c = 0 for i in range(5): if A[i] < A[9-i]: c = c + 1 k = A[i] A[i] = A[9-i] A[9-i] = k </pre>	

Ответ: _____.

19.3.2. 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	Алгоритмический язык
<pre> 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 </pre>	<pre> к := 0 нц для i от 0 до 8 если A[i] > A[i+1] то к := к + 1 t := A[i] A[i] := A[i+1] A[i+1] := t все кц </pre>
C++	Паскаль
<pre> 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; } </pre>	<pre> 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; end; </pre>
Python	
<pre> 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 </pre>	

Ответ: _____.

19.3.3. 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	Паскаль
<pre> k = 0 FOR i = 0 TO 8 IF A(i)<A(i+1) AND i MOD 2 = 0 THEN k = k + 1 t = A(i) A(i) = A(i+1) A(i+1) = t END IF NEXT i </pre>	<pre> k := 0; for i := 0 to 8 do if (A[i] < A[i+1]) and (i mod 2 = 0) then begin k := k + 1; t := A[i]; A[i] := A[i+1]; A[i+1] := t end; end; </pre>
C++	Алгоритмический язык
<pre> k = 0; for (i=0 ; i<=8 ; i++) if (A[i] < A[i+1] && i % 2 == 0) { k = k + 1; t = A[i]; A[i] = A[i+1]; A[i+1] = t; } </pre>	<pre> к := 0 нц для i от 0 до 8 если A[i]<A[i+1] и mod(i,2)=0 то к := к + 1 t := A[i] A[i] := A[i+1] A[i+1] := t все кц </pre>
Python	
<pre> k = 0 for i in range(9): if A[i] < A[i+1] and i % 2 == 0: k = k + 1 t = A[i] A[i] = A[i+1] A[i+1] = t </pre>	

Ответ: _____.

19.3.4. 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	Алгоритмический язык
<pre> 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 </pre>	<pre> к := 0 нц для i от 0 до 4 если A[2*i] >= A[2*i+1] то к := к + 1 t := A[2*i] A[2*i] := A[2*i+1] A[2*i+1] := t все кц </pre>
C++	Паскаль
<pre> 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; } </pre>	<pre> к := 0; for i := 0 to 4 do if A[2*i] >= A[2*i+1] then begin к := к + 1; t := A[2*i]; A[2*i] := A[2*i+1]; A[2*i+1] := t end; </pre>
Python	
<pre> 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 </pre>	

Ответ: _____.

19.3.5. 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	Алгоритмический язык
<pre> k = 0 FOR i = 0 TO 7 IF A(i) < A(i+2) THEN k = k + 1 t = A(i) A(i) = A(i+2) A(i+2) = t END IF NEXT i </pre>	<pre> к := 0 нц для i от 0 до 7 если A[i] < A[i+2] то к := к + 1 t := A[i] A[i] := A[i+2] A[i+2] := t все кц </pre>
C++	Паскаль
<pre> k = 0; for (i = 0 ; i <= 7 ; i++) if (A[i] < A[i+2]) { k = k + 1; t = A[i]; A[i] = A[i+2]; A[i+2] = t; } </pre>	<pre> к := 0; for i := 0 to 7 do if A[i] < A[i+2] then begin к := к + 1; t := A[i]; A[i] := A[i+2]; A[i+2] := t end; end; </pre>
Python	
<pre> k = 0 for i in range(8): if A[i] < A[i+2]: k = k + 1 t = A[i] A[i] = A[i+2] A[i+2] = t </pre>	

Ответ: _____.

19.4.1. 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	Алгоритмический язык
<pre> k = 0 FOR i = 0 TO 9 IF A(0) < A(i) THEN k = k + 1 t = A(i) A(i) = A(0) A(0) = t END IF NEXT i </pre>	<pre> k := 0 нц для i от 0 до 9 если A[0] < A[i] то k := k + 1 t := A[i] A[i] := A[0] A[0] := t все кц </pre>
C++	Паскаль
<pre> k = 0; for (i = 0 ; i <= 9 ; i++) if (A[0] < A[i]) { k = k + 1; t = A[i]; A[i] = A[0]; A[0] = t; } </pre>	<pre> k := 0; for i := 0 to 9 do if A[0] < A[i] then begin k := k + 1; t := A[i]; A[i] := A[0]; A[0] := t end; </pre>
Python	
<pre> k = 0 for i in range(10): if A[0] < A[i]: k = k + 1 t = A[i] A[i] = A[0] A[0] = t </pre>	

Ответ: _____.

19.4.2. 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	Алгоритмический язык
<pre> 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 </pre>	<pre> к := 0 нц для i от 0 до 9 если A[i] >= A[0] то к := к + 1 t := A[i] A[i] := A[0] A[0] := t все кц </pre>
C++	Паскаль
<pre> 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; } </pre>	<pre> 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; end; </pre>
Python	
<pre> 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 </pre>	

Ответ: _____.

19.4.3. В программе используется целочисленный массив А 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	Алгоритмический язык
<pre> k = 0 FOR i = 0 TO 9 IF A(i) < A(9-i) THEN k = k + 1 t = A(i) A(i) = A(9-i) A(9-i) = t END IF NEXT i </pre>	<pre> к := 0 нц для i от 0 до 9 если A[i] < A[9-i] то к := к + 1 t := A[i] A[i] := A[9-i] A[9-i] := t все кц </pre>
C++	Паскаль
<pre> k = 0; for (i = 0 ; i <= 9 ; i++) if (A[i] < A[9-i]) { k = k + 1; t = A[i]; A[i] = A[9-i]; A[9-i] = t; } </pre>	<pre> к := 0; for i := 0 to 9 do if A[i] < A[9-i] then begin к := к + 1; t := A[i]; A[i] := A[9-i]; A[9-i] := t end; end; </pre>
Python	
<pre> k = 0 for i in range(10): if A[i] < A[9-i]: k = k + 1 t = A[i] A[i] = A[9-i] A[9-i] = t </pre>	

Ответ: _____.

19.4.4. 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	Алгоритмический язык
<pre> 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 </pre>	<pre> к := 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 к := к + A[0] все кц </pre>
C++	Паскаль
<pre> 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]; } </pre>	<pre> 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; end; </pre>
Python	
<pre> 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] </pre>	

Ответ: _____.

19.4.5. 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	Алгоритмический язык
<pre> k = 10 FOR i = 0 TO 8 IF A(9-i) > A(8-i) THEN A(8-i) = A(9-i) k = k - 1 END IF NEXT i </pre>	<pre> k := 10 нц для i от 0 до 8 если A[9-i] > A[8-i] то A[8-i] := A[9-i] k := k - 1 все кц </pre>
C++	Паскаль
<pre> k = 10; for (i = 0 ; i <= 8 ; i++) if (A[9-i] > A[8-i]) { A[8-i] = A[9-i]; k = k - 1; } </pre>	<pre> k := 10; for i := 0 to 8 do if A[9-i] > A[8-i] then begin A[8-i] := A[9-i]; k := k - 1 end; end; </pre>
Python	
<pre> k = 10 for i in range(9): if A[9-i] > A[8-i]: A[8-i] = A[9-i] k = k - 1 </pre>	

Ответ: _____.

20.1. 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 4, а потом 5.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 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 </pre>	<pre> var x, L, M: integer; begin readln(x); L := 0; M := 0; 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. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, L, M; cin >> x; L = 0; M = 0; while (x > 0) { L = L + 1; if (M < x % 10) M = x % 10; x = x / 10; } cout << L << endl << M; return 0; } </pre>	<pre> алг нач цел x, L, M ввод x L := 0 M := 0 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, M кон </pre>
Python	
<pre> x = int(input()) L = 0; M = 0 while x > 0: L = L + 1 if M < x % 10: M = x % 10 x = x / 10 print (L) print (M) </pre>	

Решение. Анализируем приведённую программу. Обнаруживаем в ней цикл, который работает до тех пор, пока введённое число x больше нуля. При этом на каждом шаге цикла число x уменьшается в 10 раз (делится нацело). Узнаём в этом алгоритме алгоритм перевода натурального числа в десятичную систему счисления. А именно — делить исходное число нацело на 10, пока число не станет равно нулю. При этом что-то делать с остатками от деления на 10. В этом алгоритме на каждом шаге цикла выделяется одна десятичная цифра числа. Значит, количество шагов цикла — это количество десятичных цифр в исходном числе.

Замечаем, что переменная цикла L изначально равна нулю, а на каждом шаге цикла увеличивается на 1, т.е. в переменной L подсчитывается количество шагов цикла. Значит, количество десятичных разрядов в числе x равно 4. Ведь именно такое значение переменной L оказывается в конце работы программы.

Теперь поймём, что вычисляется в переменной M . Изначально она равна нулю. А на каждом шаге цикла с ней сравнивается остаток от деления на 10. Если остаток оказывается больше текущего значения переменной M , переменная M меняется на этот остаток. Узнаём в этих действиях алгоритм нахождения максимума. То есть: каждый элемент по очереди сравнивается с текущим значением отдельной переменной; если текущий элемент оказывается больше этой переменной, переменная становится равна текущему элементу. В качестве «текущего элемента» используется остаток от деления числа x на 10. Чуть ранее мы выяснили, что в цикле происходит перевод в десятичную систему счисления. Значит, выделение остатка от деления на 10 — это десятичные цифры числа x . Значит, в переменной M ищется максимальная десятичная цифра числа x . По условию это значение равно 5.

Получается, что нам требуется подобрать такое наименьшее число x , которое имеет 4 разряда в десятичном представлении, и при этом наибольшая цифра равна 5. Чтобы число было наименьшим, его первым разрядом имеет смысл поставить цифру 1. Чтобы в числе наибольшая цифра была 5, её имеет смысл поставить на последнюю позицию. Имеем: $1xx5$, где xx — какие-то десятичные цифры. Чтобы число было наименьшим, имеет смысл сделать эти разряды равными нулю. Получаем число 1005.

Для проверки хорошо бы убедиться в следующем:

1. Что при вводе этого числа приведённая программа действительно выдаст требуемые 4 и 5. Для этого рекомендуем аккуратно выполнить трассировку программы для найденного числа.
2. Что среди всех чисел, среди которых мы выбираем ответ (четырёхзначные, с наибольшей цифрой 5) найденное число 1005 действительно является наименьшим.

20.1.1. 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 3, а потом 7.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 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 </pre>	<pre> var x, L, M: integer; begin readln(x); L := 0; M := 0; 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. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, L, M; cin >> x; L = 0; M = 0; while (x > 0) { L = L + 1; if (M < x % 10) M = x % 10; x = x / 10; } cout << L << endl << M; return 0; } </pre>	<pre> алг нач цел x, L, M ввод x L := 0 M := 0 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, M кон </pre>
Python	
<pre> x = int(input()) M = 0; L = 0 while x > 0: L = L + 1 if M < x % 10: M = x % 10 x = x // 10 print(L) print(M) </pre>	

Ответ: _____.

20.1.2. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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, M кон </pre>
Python	
<pre> 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) </pre>	

Ответ: _____.

20.1.3. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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, M кон </pre>
Python	
<pre> 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) </pre>	

Ответ: _____.

20.1.4. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>
Python	
<pre> 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) </pre>	

Ответ: _____.

20.1.5. 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 1, а потом 60.

Бейсик	Паскаль
<pre> DIM X, M, L AS INTEGER INPUT X M = 1 L = 0 WHILE X > 0 M = M * (X MOD 10) IF X MOD 10 > 3 THEN L = L + 1 END IF X = X \ 10 WEND PRINT L PRINT M </pre>	<pre> 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 > 3 then L := L + 1; x := x div 10 end; writeln(L); write(M) end. </pre>
C++	Алгоритмический язык
<pre> #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 > 3) L = L + 1; x = x / 10; } cout << L << endl << M; return 0; } </pre>	<pre> 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) > 3 то L := L + 1 все x := div(x,10) кц вывод L, M кон </pre>
Python	
<pre> x = int(input()) M = 1; L = 0 while x > 0: M = M + x % 10 if x % 10 > 3: L = L + 1 x = x // 10 print(L) print(M) </pre>	

Ответ: _____.

Замечание к решению задач 20.2. Здесь собраны задачи, очень похожие на задачи 20.1. Они решаются очень похожим образом. Нужно только обратить внимание на то, что в процессе выделения цифр числа здесь встречается не только количество цифр в числе, но и их сумма.

20.2.1. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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, M кон </pre>

Python
<pre> 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) </pre>

Ответ: _____.

20.2.2. 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 13.

Бейсик	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>

Python
<pre> 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) </pre>

Ответ: _____.

20.2.3. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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, M кон </pre>

Python
<pre> 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) </pre>

Ответ: _____.

20.2.4. Ниже на пяти языках записан алгоритм. Получив на вход число x , этот алгоритм печатает три числа K , M и L .

Укажите наибольшее число x , при вводе которого алгоритм печатает сначала 3, потом 2, потом ещё 2.

Бейсик	Паскаль
<pre> DIM X,Y,K,L,M AS INTEGER INPUT X K = 0 L = 0 M = 0 WHILE X > 0 Y = X MOD 10 K = K + 1 IF Y MOD 3 = 0 THEN L = L + 1 END IF IF Y MOD 5 = 0 THEN M = M + 1 END IF X = X \ 10 WEND PRINT K, " ", L, " ", M </pre>	<pre> var x,y,K,L,M: integer; begin readln(x); K := 0; L := 0; M := 0; while x > 0 do begin y := x mod 10; K := K + 1; if y mod 3 = 0 then L := L + 1; if y mod 5 = 0 then M := M + 1; x := x div 10 end; writeln(K, ' ', L, ' ', M) end. </pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { int x, y, K, L, M; cin >> x; K = 0; L = 0; M = 0; while (x > 0) { y = x % 10; K = K + 1; if(y % 3 == 0) L = L + 1; if(y % 5 == 0) M = M + 1; x = x / 10; } cout << K << endl; cout << L << endl << M; return 0; }</pre>	<pre>алг нач цел x, y, K, L, M ввод x K := 0 L := 0 M := 0 нц пока x > 0 y := mod(x,10) K := K + 1 если mod(y,3) = 0 то L := L + 1 все если mod(y,5) = 0 то M := M + 1 все x := div(x,10) кц вывод K, L, M кон</pre>
Python	
<pre>x = int(input()) K = 0; L = 0; M = 0 while x > 0: y = x % 10 K = K + 1 if y % 3 == 0: L = L + 1 if y % 5 == 0: M = M + 1 x = x // 10 print(K) print(L) print(M)</pre>	

Ответ: _____.

20.2.5. Ниже на пяти языках записан алгоритм. Получив на вход число x , этот алгоритм печатает три числа K , M и L .

Укажите наименьшее число x , при вводе которого алгоритм печатает сначала 3, потом 3, потом 1.

Бейсик	Паскаль
<pre> DIM X,Y,K,L,M AS INTEGER INPUT X K = 0 L = 0 M = 0 WHILE X > 0 Y = X MOD 10 K = K + 1 IF Y MOD 2 = 0 THEN L = L + 1 END IF IF Y >= 5 THEN M = M + 1 END IF X = X \ 10 WEND PRINT K," ",L," ",M </pre>	<pre> var x,y,K,L,M: integer; begin readln(x); K := 0; L := 0; M := 0; while x > 0 do begin y := x mod 10; K := K + 1; if y mod 2 = 0 then L := L + 1; if y >= 5 then M := M + 1; x := x div 10 end; writeln(K,' ',L,' ',M) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, y, K, L, M; cin >> x; K = 0; L = 0; M = 0; while (x > 0) { y = x % 10; K = K + 1; if(y % 2 == 0) L = L + 1; if(y >= 5) M = M + 1; x = x / 10; } cout << K << endl; cout << L << endl << M; return 0; } </pre>	<pre> <u>алг</u> <u>нач</u> цел x, y, K, L, M <u>ввод</u> x K := 0 L := 0 M := 0 <u>нц</u> <u>пока</u> x > 0 y := mod(x,10) K := K + 1 <u>если</u> mod(y,2) = 0 <u>то</u> L := L + 1 <u>все</u> <u>если</u> y >= 5 <u>то</u> M := M + 1 <u>все</u> x := div(x,10) <u>кц</u> <u>вывод</u> K, <u>нс</u>, L, <u>нс</u>, M <u>кон</u> </pre>

Python
<pre> x = int(input()) K = 0; L = 0; M = 0 while x > 0: y = x % 10 K = K + 1 if y % 2 == 0: L = L + 1 if y >= 5: M = M + 1 x = x // 10 print(K) print(L) print(M) </pre>

Ответ: _____.

Замечания к решению заданий 20.3. В этом списке собраны задачи, аналогичные задаче 20.1. Но приведённые здесь алгоритмы на каждом шаге цикла делят число не на 10, а на другое число (например, 6 или 8). Это значит, что в алгоритме число переводится в соответствующую (шестеричную или восьмеричную, соответственно) систему счисления. То есть, для решения задачи нужно сначала понять, что ищет алгоритм среди этих цифр (количество, сумму, максимум, минимум, количество чётных или что-то ещё), потом подобрать такое шестеричное (или восьмеричное, или соответствующее) наибольшее/наименьшее число, которое удовлетворяет условию, а потом перевести результат в десятичную систему счисления.

20.3.1. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> x, a, b <u>ввод</u> x a := 0 b := 0 <u>нц пока</u> x > 0 <u>если</u> mod(x,2) = 0 <u>то</u> a := a + 1 <u>все</u> b := b + mod(x,6) x := div(x,6) <u>кц</u> <u>вывод</u> a, <u>нс</u>, b <u>кон</u></pre>
Python	
<pre>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)</pre>	

Ответ: _____.

20.3.2. 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 1, а потом 13.

Бейсик	Паскаль
<pre>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</pre>	<pre>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.</pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> x, a, b <u>ВВОД</u> x a := 0 b := 0 <u>нц</u> <u>пока</u> x > 0 <u>если</u> mod(x,2) <> 0 <u>то</u> a := a + 1 <u>все</u> b := b + mod(x,6) x := div(x,6) <u>кц</u> <u>ВЫВОД</u> a, <u>нс</u>, b <u>кон</u></pre>
Python	
<pre>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)</pre>	

Ответ: _____.

20.3.3. 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	Паскаль
<pre>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</pre>	<pre>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.</pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> x, L, M <u>ввод</u> x L := 0 M := 0 <u>нц</u> <u>пока</u> x > 0 M := M + 1 <u>если</u> mod(x,4) > 1 <u>то</u> L := L + 1 <u>все</u> x := div(x,4) <u>кц</u> <u>вывод</u> L, <u>нс</u>, M <u>кон</u></pre>
Python	
<pre>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)</pre>	

Ответ: _____.

20.3.4. 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 12.

Бейсик	Паскаль
<pre>DIM X, A, B AS INTEGER INPUT X A = 0 : B = 1 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</pre>	<pre>var x, a, b: integer; begin readln(x); a := 0; b := 1; 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.</pre>

C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, a, b; cin >> x; a = 0; b = 1; 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; } </pre>	<pre> алг нач цел x, a, b ввод x a := 0 b := 1 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 </pre>
Python	
<pre> x = int(input()) a = 0 b = 1 while x > 0: if x % 2 == 0: a = a + x % 8 else: b = b * (x % 8) x = x // 8 print(a) print(b) </pre>	

Ответ: _____.

20.3.5. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>
Python	
<pre> 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) </pre>	

Ответ: _____.

Замечания к решению задач 20.4. Среди этих задач собраны ещё более интересные задачи 20. Они выделены в отдельный список, чтобы вы решили не только типовые задания. Заметим, что среди используемых в этих задачах алгоритмов использованы только такие, которые либо вам хорошо знакомы, либо должны быть хорошо знакомы по программе школы.

После этих задач мы приведём краткие пояснения/подсказки по задачам на тот случай, если вам не удастся понять используемые в них алгоритмы.

Подсказка к задаче 20.4.1. Из числа 94 вычитают число x , пока не получится число, меньшее x . При этом подсчитывают количество вычитаний. Это алгоритм целочисленного деления числа 94 на x (с остатком). После деления частное и остаток упорядочиваются по возрастанию.

20.4.1. 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 7.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT X L = 94 M = 0 WHILE L >= X M = M + 1 L = L - X WEND IF M < L THEN X = M M = L L = X END IF PRINT L PRINT M </pre>	<pre> var x, L, M: integer; begin readln(x); L := 94; M := 0; while L >= x do begin M := M + 1; L := L - x end; if M < L then begin x := M; M := L; L := x end; writeln(L); write(M) end. </pre>

C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, L, M; cin >> x; L = 94; M = 0; while (L >= x) { M = M + 1; L = L - x; } if (M < L) { x = M; M = L; L = x; } cout << L << endl << M; return 0; } </pre>	<pre> алг нач цел x, L, M ввод x L := 94 M := 0 нц пока L >= x M := M + 1 L := L - x кц если M < L то x := M M := L L := x все вывод L, нс, M кон </pre>
Python	
<pre> x = int(input()) L = 94 M = 0 while L >= x: M = M + 1 L = L - x if M < L: x = M M = L L = x print(L) print(M) </pre>	

Ответ: _____.

Подсказка к задаче 20.4.2. К числу ноль x раз прибавляют число y . Это алгоритм умножения x на y . Так как результат получается чётный, то и одно из чисел (x или y) должно быть чётным. Это позволяет понять, какой должен быть y .

20.4.2. Ниже на пяти языках записан алгоритм. Получив на вход число x , этот алгоритм печатает число M . Укажите число x , при вводе которого алгоритм печатает 126.

Бейсик	Паскаль
<pre> DIM X, Y, L, M AS INTEGER INPUT X L = 0 : M = 0 Y = 7 IF X MOD 2 = 0 THEN Y = 9 END IF WHILE L < X M = M + y L = L + 1 WEND PRINT M </pre>	<pre> var x, y, L, M: integer; begin readln(x); L := 0; M := 0; y := 7; if x mod 2 = 0 then y := 9; while L < x do begin M := M + y; L := L + 1 end; write(M) end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { int x, y, L, M; cin >> x; L = 0; M = 0; y = 7; if (x % 2 == 0) y = 9; while (L < x) { M = M + y; L = L + 1; } cout << M << endl; return 0; } </pre>	<pre> алг нач цел x, y, L, M ввод x L := 0; M := 0 y := 7 если mod(x,2)=0 то y := 9 все нц пока L < x M := M + y L := L + 1 кц вывод M кон </pre>
Python	
<pre> x = int(input()) L = 0; M = 0; y = 7 if x % 2 == 0: y = 9 while L < x: M = M + y L = L + 1 print(M) </pre>	

Ответ: _____.

Подсказка к задаче 20.4.3. Пока число M не равно нулю, делаем его равным остатку от деления числа L на M и меняем числа местами. Это ускоренный алгоритм Евклида (нахождение наибольшего общего делителя чисел L и M). Так как результат равен 6, а исходное число M равно 32 (не кратно 6), делаем ввод, что M не 32, а 36. Значит, L (и x) должно делиться на 7. Остаётся подобрать наименьшее число, кратное 6 и 7, и имеющее $\text{НОД}(x, 36) = 6$.

20.4.3. 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	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>

Python
<pre> 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) </pre>

Ответ: _____.

Подсказка к задаче 20.4.4. В цикле пока M не равно L вычитаем из большего числа меньшее. Это алгоритм Евклида. Так как результат алгоритма (НОД) равен 14, оба числа должны быть кратны 14. Значит, L — чётное число. Значит, $M = 28$. Найдём такое число, которое больше 50 и делится на 14. Наименьшее — 56. Но $\text{НОД}(56, 28) = 28$. Значит, нужно взять следующее число и проверить его.

20.4.4. 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	Паскаль
<pre> 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 </pre>	<pre> 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; end if; end while; write(M) end. </pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> цел x, L, M <u>ввод</u> x M := 42; L := x <u>если</u> mod(L, 2) = 0 <u>то</u> M := 28 <u>все</u> <u>нц пока</u> L <> M <u>если</u> L > M <u>то</u> L := L - M <u>иначе</u> M := M - L <u>все</u> <u>кц</u> <u>вывод</u> M <u>кон</u></pre>
Python	
<pre>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)</pre>	

Ответ: _____.

Подсказка к задаче 20.4.5. Пока x больше нуля, делим его на 2. Это перевод в двоичную систему счисления. Значит, M — количество разрядов, L — количество нулей.

20.4.5. Ниже на пяти языках записан алгоритм. Получив на вход число x , этот алгоритм печатает два числа L и M . Укажите наибольшее число x , при вводе которого алгоритм печатает сначала 3, а потом 7.

Бейсик	Паскаль
<pre> DIM X, L, M AS INTEGER INPUT X L = 0 M = 0 WHILE X > 0 M = M + 1 IF X MOD 2 = 0 THEN L = L + 1 END IF X = X \ 2 WEND PRINT L PRINT M </pre>	<pre> var x, L, M: integer; begin readln(x); L := 0; M := 0; while x > 0 do begin M := M + 1; if x mod 2 = 0 then L := L + 1; x := x div 2 end; writeln(L); write(M) end. </pre>
C++	Алгоритмический язык
<pre> #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 % 2 == 0) L = L + 1; x = x / 2; } cout << L << endl << M; return 0; } </pre>	<pre> алг нач цел x, L, M ввод x L := 0 M := 0 нц пока x > 0 M := M + 1 если mod(x,2) = 0 то L := L + 1 все x := div(x,2) кц вывод L, нс, M кон </pre>
Python	
<pre> x = int(input()) L = 0 M = 0 while x > 0: M = M + 1 if x % 2 == 0: L = L + 1 x = x // 2 print(L) print(M) </pre>	

Ответ: _____.

21.1. Ниже на пяти языках представлен алгоритм. Определите, при скольких значениях b на экран будет выведено такое же число, как и при $b = 100$ (включая $b = 100$).

Бейсик	Паскаль
<pre> 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=6+F(x-1) END IF END FUNCTION </pre>	<pre> var a, b : integer; function F(x:integer):integer; begin if x = 0 then F := 0 else F := 6 + F(x - 1) end; BEGIN readln(b); a := 0; while F(a) < b do a := a + 1; write(a) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { if(x == 0) return 0; else return 6+F(x-1); } int main() { long a, b; cin >> b; a = 0; while (F(a) < b) a = a + 1; cout << a << endl; return 0; } </pre>	<pre> 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 := 6 + F(x - 1) все кон </pre>

Python

```
def F(x):  
    if x == 0:  
        return 0  
    else:  
        return 6 + F(x - 1)  
b = int(input())  
a = 0  
while F(a) < b:  
    a = a + 1  
print(a)
```

Решение. Сначала определим, что вычисляет рекуррентная функция. При каждом рекуррентном вызове значение аргумента уменьшается на 1, пока не станет равно нулю. При этом на каждом шаге функция вызывается один раз. Значит, функция $F(x)$ вызовется x раз. Кроме этого, на каждом шаге значение функции увеличивается на 6. То есть, x раз прибавляется 6. Значит, $F(x) = 6x$.

Теперь исследуем основную программу. Найдём, что выведет программа при вводе числа 100. Видим цикл с предусловием `while`, в котором переменная a на каждом шаге увеличивается на 1 до тех пор, пока $F(a) < b$. При некотором значении переменной a цикл заканчивается. То есть, на экран будет выведено наименьшее целое число a , при котором перестанет выполняться условие $F(a) < b$. То есть, начнёт выполняться условие $F(a) \geq b$. Подставим в это неравенство значения для $F(a)$ и для b : $6a \geq 100 \Rightarrow a \geq 100/6 \Rightarrow a \geq 16 \frac{4}{6}$. Так как мы ищем наименьшее целое значение a , то $a = 17$. Итак, мы нашли, что на экран будет выведено число 17.

Теперь найдём, при каких значениях переменной b это будет так. Так как цикл `while F(a) < b` выполняется шаг за шагом, от начального значения $a = 0$, увеличивая значение a на 1, получаем, что:

при $a = 16$ цикл ещё выполняется. То есть, $F(16) < b$;

при $a = 17$ цикл заканчивается. То есть, $F(17) \geq b$.

Подставляем числа 16 и 17 в полученную нами формулу для $F(x)$. Получаем: $6 \cdot 16 < b \leq 6 \cdot 17 \Rightarrow 96 < b \leq 102$. Подсчитываем количество целых значений переменной b , удовлетворяющее этому условию: $102 - 96 = 6$.

Если эта разность для вас не очевидна, просто посчитайте все подряд идущие подходящие значения: 97, 98, 99, 100, 101, 102. Всего 6 штук.

Ответ: 6.

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

Бейсик	Паскаль
<pre> DIM A, B AS INTEGER INPUT B A = 0 WHILE F(A) < B A = A + 1 WEND PRINT A FUNCTION F (x) DIM I, S AS INTEGER S = 0 I = 1 WHILE I <= x S = S + 5 I = I + 1 WEND F = S END FUNCTION </pre>	<pre> var a,b : integer; function F(x:integer):integer; var i, s : integer; begin s := 0; i := 1; while i <= x do begin s := s + 5; i := i + 1 end; F := s end; BEGIN readln(b); a := 0; while F(a) < b do a := a + 1; write(a) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { long i, s; s = 0; i = 1; while(i <= x) { s = s + 5; i = i + 1; } return s; } int main() { int a, b; cin >> b; a = 0; while (F(a) < b) a = a + 1; cout << a << endl; return 0; } </pre>	<pre> 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 цел i, s s := 0; i := 1 нц пока i <= x s := s + 5 i := i + 1 кц знач := s кон </pre>

Python
<pre>def F(x): s = 0 i = 1 while i <= x: s = s + 5 i = i + 1 return s b = int(input()) a = 0 while F(a) < b: a = a + 1 print(a)</pre>

Ответ: _____.

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

Бейсик	Паскаль
<pre>DIM A, B AS INTEGER INPUT B A = 0 WHILE F(A) < B A = A + 1 WEND PRINT A FUNCTION F (x) DIM I, S AS INTEGER S = 0 I = 0 WHILE I < x S = S + 10 I = I + 1 WEND F = S END FUNCTION</pre>	<pre>var a,b : integer; function F(x:integer):integer; var i, s : integer; begin s := 0; i := 0; while i < x do begin s := s + 10; i := i + 1 end; F := s end; BEGIN readln(b); a := 0; while F(a) < b do a := a + 1; write(a) END.</pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; long F(long x) { long i, s; s = 0; i = 0; while(i < x) { s = s + 10; i = i + 1; } return s; } int main() { long a, b; cin >> b; a = 0; while (F(a) < b) a = a + 1; cout << a << endl; return 0; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> a, b <u>ввод</u> b a := 0 <u>нц пока</u> F(a) < b a := a + 1 <u>кц</u> <u>вывод</u> a <u>кон</u> <u>алг цел</u> F(<u>цел</u> x) <u>нач</u> <u>цел</u> i, s s := 0 i := 0 <u>нц пока</u> i < x s := s + 10 i := i + 1 <u>кц</u> <u>знач</u> := s <u>кон</u></pre>
Python	
<pre>def F(x): s = 0 i = 0 while i < x: s = s + 10 i = i + 1 return s b = int(input()) a = 0 while F(a) < b: a = a + 1 print(a)</pre>	

Ответ: _____.

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

Бейсик	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>
Python	
<pre> 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) </pre>	

Ответ: _____.

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

Бейсик	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>

Python
<pre>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)</pre>

Ответ: _____.

21.1.5. Ниже на пяти языках представлен алгоритм. Определите, при скольких значениях b на экран будет выведено такое же число, как и при $b = 100$ (включая $b = 100$).

Бейсик	Паскаль
<pre>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 = 1 ELSE F = 2 * F(x - 1) END IF END FUNCTION</pre>	<pre>var a,b : integer; function F(x:integer):integer; begin if x = 0 then F := 1 else F := 2 * F(x - 1) end; BEGIN readln(b); a := 0; while F(a) < b do a := a + 1; write(a) END.</pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; long F(long x) { if(x == 0) return 1; else return 2 * F(x-1); } int main() { long a, b; cin >> b; a = 0; while (F(a) < b) a = a + 1; cout << a << endl; return 0; }</pre>	<pre>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 := 1 иначе знач := 2 * F(x - 1) все кон</pre>
Python	
<pre>def F(x): if x == 0: return 1 else: return 2 * F(x - 1) b = int(input()) a = 0 while F(a) < b: a = a + 1 print(a)</pre>	

Ответ: _____.

Замечание к решению заданий 21.2. Эти задания решаются аналогично задаче 21.1. Сначала желательно определить, что вычисляет функция. Потом подставить данное число в качестве входного значения и определить, какое число программа выводит на экран. Потом проанализировать, при каком наименьшем и/или наибольшем значении вводимой переменной (в зависимости от вопроса задачи) цикл заканчивается на этом же шаге. Эти задания выделены в отдельный блок, чтобы сформировать у вас более устойчивый навык решения подобных заданий.

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

Бейсик	Паскаль
<pre> 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 </pre>	<pre> 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. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>

Python
<pre>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)</pre>

Ответ: _____.

21.2.2. Определите, при каком наибольшем значении b в результате выполнения следующего алгоритма будет напечатано число 10 (для Вашего удобства алгоритм представлен на пяти языках):

Бейсик	Паскаль
<pre>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</pre>	<pre>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.</pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> a, b <u>ввод</u> b a := 0 <u>нц пока</u> a < F(b) a := a + 1 <u>кц</u> <u>вывод</u> a <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>если</u> x < 7 <u>то</u> <u>знач</u> := 0 <u>иначе</u> <u>знач</u> := F(x - 7) + 1 <u>все</u> <u>кон</u></pre>
Python	
<pre>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)</pre>	

Ответ: _____.

21.2.3. Ниже на пяти языках представлен алгоритм. Определите, при скольких значениях b на экран будет выведено такое же число, как и при $b = 200$ (включая $b = 200$).

Бейсик	Паскаль
<pre>DIM A, B AS INTEGER INPUT B A = 0 WHILE A < F(B) A = A + 1 WEND PRINT A</pre>	<pre>var a,b : integer; function F(x:integer):integer; begin if x = 0 then F := 0 else F := F(x div 2) + 1 end;</pre>

<pre> FUNCTION F (x) IF x = 0 F = 0 ELSE F = F(x \ 2) + 1 END IF END FUNCTION </pre>	<pre> BEGIN readln(b); a := 0; while a < F(b) do begin a := a + 1 end; write(a) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(int x) { if(x == 0) return 0; else return F(x/2) + 1; } int main() { int a, b; cin >> b; a = 0; while (a < F(b)) a = a + 1; cout << a << endl; return 0; } </pre>	<pre> 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 := 0 иначе знач := F(div(x,2)) + 1 кц кон </pre>
Python	
<pre> def F(x): if x == 0: return 0 else: return F(x // 2) + 1 b = int(input()) a = 0 while a < F(b): a = a + 1 print(a) </pre>	

Ответ: _____.

21.2.4. Ниже на пяти языках представлен алгоритм. Определите, при каком наибольшем значении b в результате выполнения следующего алгоритма будет напечатано такое же число, как и при $b = 300$:

Бейсик	Паскаль
<pre> 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 = 0 ELSE F = F(x\2) + 1 END IF END FUNCTION </pre>	<pre> var a,b : integer; function F(x:integer):integer; begin if x = 0 then F := 0 else F := F(x div 2) + 1 end; BEGIN readln(b); a := 0; while a < F(b) do begin a := a + 1 end; write(a) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(int x) { if(x == 0) return 0; else return F(x/2) + 1; } int main() { int a, b; cin >> b; a = 0; while (a < F(b)) a = a + 1; cout << a << endl; return 0; } </pre>	<pre> 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 := 0 иначе знач := F(div(x,2)) + 1 кц кон </pre>
Python	
<pre> def F(x): if x == 0: return 0 else: return F(x // 2) + 1 b = int(input()) a = 0 while a < F(b): a = a + 1 print(a) </pre>	

Ответ: _____.

21.2.5. Ниже на пяти языках представлен алгоритм. Определите, при скольких значениях b на экран будет выведено такое же число, как и при $b = 300$ (включая $b = 300$).

Бейсик	Паскаль
<pre> 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 = 1 ELSE F = 2 * F(x-1) END IF END FUNCTION </pre>	<pre> var a,b : integer; function F(x:integer):integer; begin if x = 0 then F := 1 else F := 2 * F(x - 1) end; BEGIN readln(b); a := 0; while F(a) < b do a := a + 1; write(a) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { if(x == 0) return 1; else return 2 * F(x-1); } int main() { long a, b; cin >> b; a = 0; while (F(a) < b) a = a + 1; cout << a << endl; return 0; } </pre>	<pre> 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 := 1 иначе знач := 2 * F(x - 1) все кон </pre>
Python	
<pre> def F(x): if x == 0: return 1 else: return 2 * F(x - 1) b = int(input()) a = 0 while F(a) < b: a = a + 1 print(a) </pre>	

Ответ: _____.

21.3. 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 $k = 20$. Значение $k = 20$ также включается в подсчет различных значений k .

Бейсик	Паскаль
<pre> 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 - 1 ELSE PRINT I END IF FUNCTION F (n) F = n * n - 5 END FUNCTION </pre>	<pre> var k, i : longint; function f(n:longint):longint; begin f := n * n - 5 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 - 1) else writeln(i) end. end. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long f(long n) { return n * n - 5; } int main() { long k, i; cin >> k; i = 0; while (f(i) < k) i++; if (f(i)+f(i-1)>2*k) cout << i-1 << endl; else cout << i << endl; return 0; } </pre>	<pre> 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 - 1 иначе вывод i все кон алг цел F(цел n) нач знач := n * n - 5 кон </pre>

Python

```
def f(n):  
    return n * n - 5  
k = int(input())  
i = 0  
while f(i) < k:  
    i = i + 1  
if f(i) + f(i - 1) > 2 * k:  
    print(i - 1)  
else:  
    print(i)
```

Решение.

Выясним, какое число выводит на экран программа при $k=20$.

Сначала определим, при каком значении переменной i остановится цикл `while`. Это случится при наименьшем i , при котором перестанет выполняться условие цикла. То есть, при котором $f(i) \geq k$. Подставим в неравенство значения: $i \cdot i - 5 \geq 20 \Rightarrow i^2 \geq 25 \Rightarrow |i| \geq 5$. Так как i начинается от 0 и увеличивается на 1, наименьшее i , при котором это выполнится, будет 5.

Подставим это значение в условие: $f(5) + f(4) > 2 \cdot 20 \Rightarrow 20 + 11 > 40$. Неверно. Значит, на экран выведется i , то есть, 5. Теперь найдём, при каких k на экране будет 5. Это может случиться при выводе на экран $i=5$, или при выводе на экран $i-1=5$.

Случай 1. Если 5 выведено на экран через вывод $i=5$. Значит, условие не выполнилось. То есть, $f(5) + f(5-1) \leq 2k \Rightarrow 20 + 11 \leq 2k \Rightarrow 31 \leq 2k \Rightarrow k \geq 15,5$.

Кроме того, цикл закончился при $i=5$. Значит, при $i=4$ цикл ещё продолжался и условие цикла выполнялось: $f(4) < k \Rightarrow 11 < k$, а при $i=5$ цикл закончился и условие цикла перестало выполняться: $f(5) \geq k \Rightarrow 20 \geq k$.

Объединяем (в систему) все три полученных неравенства:

$$k \geq 15,5$$

$$11 < k$$

$$20 \geq k$$

Получаем: $15,5 \leq k \leq 20$. Так как k — целое, можно записать это как $16 \leq k \leq 20$.

Случай 2. Если 5 выведено на экран через вывод $i-1=5$, значит условие выполнилось. То есть, $f(6)+f(6-1)>2k \Rightarrow \Rightarrow 31+20>2k \Rightarrow 51>2k \Rightarrow k<25,5$.

Кроме того, цикл закончился при $i=6$. Значит, при $i=5$ цикл ещё продолжался и условие цикла выполнялось: $f(5)<k \Rightarrow \Rightarrow 20<k$, а при $i=6$ цикл закончился и условие цикла перестало выполняться: $f(6)\geq k \Rightarrow 31\geq k$.

Объединяем (в систему) все три полученных неравенства:

$$k<25,5$$

$$20<k$$

$$31\geq k$$

Получаем: $20<k<25,5$. Так как k — целое, можно записать это как $20\leq k\leq 25$.

Объединяем неравенства из случаев 1 и 2. Получаем: $16\leq k\leq 25$.

Подсчитаем количество целых k , удовлетворяющих этому неравенству: $25 - 16 + 1 = 10$ (а именно: 16, 17, 18, 19, 20, 21, 22, 23, 24, 25).

Ответ: 10.

21.3.1. Ниже на пяти языках представлен алгоритм. Напишите в ответе наименьшее значение входной переменной k , при котором программа выдаёт тот же ответ, что и при входном значении $k = 23$.

Бейсик	Паскаль
<pre> DIM K, A AS INTEGER INPUT K A = 1 WHILE F(A) < K A = A + 1 WEND PRINT A FUNCTION F (n) F = n * n + 3 END FUNCTION </pre>	<pre> var k, a : integer; function f(n:integer):integer; begin f := n * n + 3 end; BEGIN readln(k); a := 1; while f(a) < k do a := a + 1; write(a) END. </pre>

Си	Алгоритмический язык
<pre>#include <iostream> using namespace std; long f(long n) { return n * n + 3; } int main() { long k, a; cin >> k; a = 1; while (f(a) < k) a = a + 1; cout << a << endl; return 0; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> k, a <u>ввод</u> k a := 1 <u>нц</u> <u>пока</u> F(a) < k a := a + 1 <u>кц</u> <u>вывод</u> a <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> n) <u>нач</u> <u>знач</u> := n * n + 3 <u>кон</u></pre>
Python	
<pre>def F(x): return n * n + 3 k = int(input()) a = 1 while F(a) < k: a = a + 1 print(a)</pre>	

Ответ: _____.

21.3.2. Ниже на пяти языках представлен алгоритм. Напишите в ответе наименьшее значение входной переменной k , при котором программа выдаёт тот же ответ, что и при входном значении $k = 22$.

Бейсик	Паскаль
<pre>DIM K, A AS INTEGER INPUT K A = 1 WHILE F(A) < G(K) A = A + 1 WEND PRINT A FUNCTION F (n) F = n * n - 5 END FUNCTION FUNCTION G (n) G = 3 * n + 8 END FUNCTION</pre>	<pre>var k, a : integer; function f(n:integer):integer; begin f := n * n - 5 end; function g(n:integer):integer; begin g := 3 * n + 8 end; BEGIN readln(k); a := 1; while f(a) < g(k) do a := a + 1; write(a) END.</pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; #include <stdio.h> long f(long n) { return n * n - 5; } long g(long n) { return 3 * n + 8; } int main() { long k, a; cin >> k; a = 1; while (f(a) < g(k)) a = a + 1; cout << a << endl; return 0; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> k, a <u>ввод</u> k a := 1 <u>нц</u> <u>пока</u> F(a) < G(k) a := a + 1 <u>кц</u> <u>вывод</u> a <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> n) <u>нач</u> <u>знач</u> := n * n - 5 <u>кон</u> <u>алг</u> <u>цел</u> G(<u>цел</u> n) <u>нач</u> <u>знач</u> := 3 * n + 8 <u>кон</u></pre>
Python	
<pre>def f(n): return n * n - 5 def g(n): return 3 * n + 8 k = int(input()) a = 1 while f(a) < g(k): a = a + 1 print(a)</pre>	

Ответ: _____.

21.3.3. Ниже на пяти языках представлен алгоритм. Напишите в ответе наименьшее значение входной переменной k , при котором программа выдаёт тот же ответ, что и при входном значении $k = 64$.

Бейсик	Паскаль
<pre> DIM K, A AS INTEGER INPUT K A = 1 WHILE F(A) < G(A,K) A = A + 1 WEND PRINT A FUNCTION F (n) F = n * n - 15 END FUNCTION FUNCTION G (n,k) G = 2 * n + k END FUNCTION </pre>	<pre> var k, a : integer; function f(n:integer):integer; begin f := n * n - 15 end; function g(n,k:integer):integer; begin g := 2 * n + k end; BEGIN readln(k); a := 1; while f(a) < g(a,k) do a := a + 1; write(a) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; #include <stdio.h> long f(long n) { return n * n - 15; } long g(long n, long k) { return 2 * n + k; } int main(){ long k, a; cin >> k; a = 1; while (f(a) < g(a,k)) a = a + 1; cout << a << endl; return 0; } </pre>	<pre> алг нач цел k, a ввод k a := 1 нц пока F(a) < G(a,k) a := a + 1 кц вывод a кон алг цел F(цел n) нач знач := n * n - 15 кон алг цел G(цел n) нач знач := 2 * n + k кон </pre>
Python	
<pre> def f(n): return n * n - 15 def g(n, k): return 2 * n + k k = int(input()) a = 1 while f(a) < g(a,k): a = a + 1 print(a) </pre>	

Ответ: _____.

21.3.4. 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 $k = 40$. Значение $k = 40$ также включается в подсчёт различных значений k .

Бейсик	Паскаль
<pre> 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 </pre>	<pre> 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. end. </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>

Python
<pre>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)</pre>

Ответ: _____.

21.3.5. 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 подсчёт различных значений k .

Бейсик	Паскаль
<pre>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</pre>	<pre>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. end.</pre>

C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>
Python	
<pre> 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) </pre>	

Ответ: _____.

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

Бейсик	Паскаль
<pre> DIM A,B,T,M,R AS INTEGER A = -40: B = 40 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 + 8 FUNCTION F (x) F=(x*x-36)*(x*x-36)+7 END FUNCTION </pre>	<pre> var a,b,t,M,R : integer; function F(x:integer):integer; begin F:=(x*x-36)*(x*x-36)+7 end; BEGIN a := -40; b := 40; 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 + 8) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { return (x*x-36)*(x*x-36)+7; } int main() { long a, b, t, M, R; a = -40; b = 40; M = a; R = F(a); for (t=a; t<=b; t++) if (F(t) < R) { M = t; R = F(t); } cout << M+8 << endl; return 0; } </pre>	<pre> алг нач цел a, b, t, M, R a := -40; b := 40 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 + 8 кон алг цел F(цел x) нач знач:=(x*x-36)*(x*x-36)+7 кон </pre>

Python

```
def F(x):  
    return (x*x - 36)*(x*x - 36) + 7  
a = -40; b = 40  
M = a; R = F(a)  
for t in range(a,b+1):  
    if (F(t) < R):  
        M = t; R = F(t)  
print(M + 8)
```

Решение.

Анализируем программу. Видим, что в ней перебираются целочисленные значения переменной t от -40 до 40 . 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 $[-40, 40]$ среди целочисленных значений аргумента.

Анализируем функцию. Понимаем, что она сводится к виду: $F(x) = (x^2 - 6^2)^2 + 7$. Понимаем, что минимум функции равен 7 . И что это значение происходит при $(x^2 - 6^2)^2 = 0$. Отсюда находим значения переменной x , при которых это выполняется: $x^2 - 6^2 = 0 \Rightarrow \Rightarrow x^2 = 6^2 \Rightarrow |x| = 6 \Rightarrow x = \pm 6$.

Среди двух значений нам нужно выбрать то, которое окажется в переменной M . Значения аргумента функции перебираются последовательно от -40 до 40 . Соответственно, первым из чисел -6 и 6 будет найдено число -6 . Так как при поиске минимума используется строгое неравенство, при значении аргумента, равном 6 , условие $F(t) < R$ не выполнится (потому что в этот момент $F(t)$ будет равно R). Поэтому переменная M не изменится и останется равна -6 до конца работы цикла.

На экран выводится результат выражения $(M+8)$. Подставим в него значение -6 .

Ответ: 2.

❗ Если бы при поиске минимума использовалось нестрогое неравенство ($F(t) \leq R$), то для $t=6$ условие бы тоже выполнилось, и в переменной M оказалось бы число 6 .

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

Бейсик	Паскаль
<pre> DIM A,B,T,M,R AS INTEGER A = -25: B = 25 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 + 9 FUNCTION F (x) F=(x*x-9)*(x*x-9)-5 END FUNCTION </pre>	<pre> var a,b,t,M,R : integer; function F(x:integer):integer; begin F := (x*x-9)*(x*x-9)-5 end; BEGIN a := -25; b := 25; 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 + 9) END. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { return (x*x-9)*(x*x-9)-5; } int main() { long a, b, t, M, R; a = -25; b = 25; M = a; R = F(a); for (t=a; t<=b; t++) if (F(t)<R) { M = t; R = F(t); } cout << M+9 << endl; return 0; } </pre>	<pre> алг нач цел a, b, t, M, R a := -25; b := 25 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 + 9 кон алг цел F(цел x) нач знач := (x*x-9)*(x*x-9)-5 кон </pre>

Python
<pre>def F(x): return (x*x-9)*(x*x-9)-5 a = -25 b = 25 M = a R = F(a) for t in range(a,b+1): if (F(t) < R): M = t R = F(t) print(M + 9)</pre>

Ответ: _____.

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

Бейсик	Паскаль
<pre>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 + 6 FUNCTION F (x) F = (x*x-16)*(x*x-16)+3 END FUNCTION</pre>	<pre>var a,b,t,M,R : integer; function F(x:integer):integer; begin F:=(x*x-16)*(x*x-16)+3 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 + 6) END.</pre>

C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; long F(long x) { return (x*x-16)*(x*x-16)+3; } 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+6 << endl; return 0; } </pre>	<pre> 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 + 6 кон алг цел F(цел x) нач знач:=(x*x-16)*(x*x-16)+3 кон </pre>
Python	
<pre> def F(x): return (x*x - 16)*(x*x - 16) + 3 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 + 6) </pre>	

Ответ: _____.

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

Бейсик	Паскаль
<pre> DIM A,B,T,M,R AS INTEGER A = -25: B = 25 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)+5 END FUNCTION </pre>	<pre> var a,b,t,M,R : integer; function F(x:integer):integer; begin F:=(x*x-4) * (x*x-4)+5 end; BEGIN a := -25; b := 25; M := a; R := F(a); for t := a to b do begin if F(t)<R then begin M := t; R := F(t) end end; write(M + R) END. </pre>
C++	Алгоритмический язык
<pre> int F(int x) { return (x*x-4)*(x*x-4)+5; } int main() { int a, b, t, M, R; a = -25; b = 25; M = a; R = F(a); for (t=a; t<=b; t++) { if (F(t)<R) { M = t; R = F(t); } } std::cout << M+R; } </pre>	<pre> алг нач цел a, b, t, M, R a := -25; b := 25 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)+5 кон </pre>

Python
<pre>def F(x): return (x*x-4)*(x*x-4)+5 a = -25 b = 25 M = a R = F(a) for t in range(a,b+1): if (F(t)<R): M = t; R = F(t) print(M + R)</pre>

Ответ: _____.

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

Бейсик	Паскаль
<pre>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</pre>	<pre>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.</pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> a, b, t, M, R a := -30; b := 30 M := a; R := F(a) <u>нц</u> <u>для</u> t <u>от</u> a <u>до</u> b <u>если</u> F(t) <= R <u>то</u> M := t R := F(t) <u>все</u> <u>кц</u> <u>вывод</u> M + R <u>кон</u> <u>алг</u> <u>цел</u> F(<u>цел</u> x) <u>нач</u> <u>знач</u> := (x*x-25) * (x*x-25) + 8 <u>кон</u></pre>
Python	
<pre>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)</pre>	

Ответ: _____.

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

Бейсик	Паскаль
<pre> DIM A,B,T,M,R AS INTEGER A = -25: B = 25 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=abs(abs(x-4)+abs(x+4)-10)+9 END FUNCTION </pre>	<pre> var a,b,t,M,R : integer; function F(x:integer):integer; begin F:=abs(abs(x-4)+abs(x+4)-10)+9 end; BEGIN a := -25; b := 25; 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. </pre>
C++	Алгоритмический язык
<pre> #include <iostream> #include <cstdlib> using namespace std; long F(long x) { return abs(abs(x-4)+ abs(x+4)-10)+9; } int main() { int a, b, t, M, R; a = -25; b = 25; 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; } </pre>	<pre> алг нач цел a, b, t, M, R a := -25; b := 25 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:=abs(abs(x-4)+abs(x+4)-10)+9 кон </pre>
Python	
<pre> def F(x): return abs(abs(x-4)+abs(x+4)-10)+9 a = -25; b = 25 M = a; R = F(a) for t in range(a,b+1): if (F(t)<R): M = t; R = F(t) print(M + R) </pre>	

Ответ: _____.

22.1. 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 3 преобразуют в число 23?

Решение.

Способ 1. Обозначим как $F(n)$ количество способов получить из числа 3 число n .

Так как доступные действия (+1 и $\times 2$) не могут уменьшить целое число, можем сделать вывод, что $F(n) = 0$ для $n < 3$ (число программ, которыми можно получить из числа 3 число, меньшее трёх, равно нулю).

Кроме того, важно понимать, что $F(3) = 1$. То есть, количество различных программ, которыми можно получить из числа 3 само число 3, равно одному — пустая программа.

Теперь составим функцию для вычисления $F(n)$ при $n > 3$.

Так как возможными командами являются добавление единицы и удвоение, можно сделать вывод, что нечётное число может быть получено только прибавлением единицы к числу на 1 меньше. То есть, $F(n) = F(n-1)$, если n — нечётное > 3 .

Для чётного числа n чуть сложнее — оно может быть получено как прибавлением единицы к предыдущему числу ($n-1+1$), так и удвоением числа, вдвое меньшего n ($n/2 \times 2$).

То есть, $F(n) = F(n-1) + F(n/2)$, если n — чётное > 3 .

Запишем полученные формулы все вместе:

$F(n) = 0$, если $n < 3$,

$F(n) = 1$, если $n = 3$,

$F(n) = F(n-1)$, если n — нечётное, $n > 3$,

$F(n) = F(n-1) + F(n/2)$, если n — чётное, $n > 3$.

Будем вычислять последовательно значения функции от $F(4)$ до $F(23)$:

$F(4) = F(4-1) + F(4/2) = F(3) + F(2) = 1 + 0 = 1$

$F(5) = F(5-1) = F(4) = 1$

$$F(6) = F(6-1) + F(6/2) = F(5) + F(3) = 1 + 1 = 2$$

$$F(7) = F(7-1) = F(6) = 2$$

$$F(8) = F(8-1) + F(8/2) = F(7) + F(4) = 2 + 1 = 3$$

$$F(9) = F(9-1) = F(8) = 3$$

$$F(10) = F(10-1) + F(10/2) = F(9) + F(5) = 3 + 1 = 4$$

$$F(11) = F(11-1) = F(10) = 4$$

$$F(12) = F(12-1) + F(12/2) = F(11) + F(6) = 4 + 2 = 6$$

$$F(13) = F(13-1) = F(12) = 6$$

$$F(14) = F(14-1) + F(14/2) = F(13) + F(7) = 6 + 2 = 8$$

$$F(15) = F(15-1) = F(14) = 8$$

$$F(16) = F(16-1) + F(16/2) = F(15) + F(8) = 8 + 3 = 11$$

$$F(17) = F(17-1) = F(16) = 11$$

$$F(18) = F(18-1) + F(18/2) = F(17) + F(9) = 11 + 3 = 14$$

$$F(19) = F(19-1) = F(18) = 14$$

$$F(20) = F(20-1) + F(20/2) = F(19) + F(10) = 14 + 4 = 18$$

$$F(21) = F(21-1) = F(20) = 18$$

$$F(22) = F(22-1) + F(22/2) = F(21) + F(11) = 18 + 4 = 22$$

$$F(23) = F(23-1) = F(22) = 22$$

Более удобно (хотя и не так подробно и понятно) те же вычисления можно записать в виде таблицы:

n	3	4	5	6	7	8	9	10	11	12	13	14
F(n)	1	1	1	2	2	3	3	4	4	6	6	8

Окончание табл.

n	15	16	17	18	19	20	21	22	23
F(n)	8	11	11	14	14	18	18	22	22

Ответ: 22.

22.1.1. 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: _____.

22.1.2. 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: _____.

22.1.3. 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: _____.

22.1.4. 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 1 преобразуют в число 29?

Ответ: _____.

22.1.5. У исполнителя Утроитель две команды, которым присвоены номера:

1. прибавь 2,

2. умножь на 3.

Первая из них увеличивает число на экране на 2, вторая утраивает его.

Программа для Утроителя — это последовательность команд.

Сколько есть программ, которые число 2 преобразуют в число 28?

Ответ: _____.

Замечание к решению заданий 22.2. В этой группе собраны задания, очень похожие на задания группы 22.1. Но в них вместо команды умножения используется ещё одна или две команды сложения. Соответственно, решение этих заданий совершенно аналогично решению задания 22.1. Даже проще. В рекуррентной формуле не нужно будет различать случаи кратности/некратности $2/3$.

22.2.1. 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: _____.

22.2.2. 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: _____.

22.2.3. У исполнителя Прибавлятель три команды, которым присвоены номера:

1. прибавь 1,
2. прибавь 2,
3. прибавь 5.

Первая из них увеличивает число на экране на 1, вторая — на 2, третья — на 5.

Программа для Прибавлятеля — это последовательность команд. Сколько есть программ, которые число 2 преобразуют в число 10?

Ответ: _____.

22.2.4. У исполнителя Прибавлятель три команды, которым присвоены номера:

- 1. прибавь 1,**
- 2. прибавь 2,**
- 3. прибавь 4.**

Первая из них увеличивает число на экране на 1, вторая — на 2, третья — на 4.

Программа для Прибавлятеля — это последовательность команд. Сколько есть программ, которые число 1 преобразуют в число 11?

Ответ: _____.

22.2.5. У исполнителя Прибавлятель три команды, которым присвоены номера:

- 1. прибавь 1,**
- 2. прибавь 4,**
- 3. прибавь 5.**

Первая из них увеличивает число на экране на 1, вторая — на 4, третья — на 5.

Программа для Прибавлятеля — это последовательность команд. Сколько есть программ, которые число 1 преобразуют в число 15?

Ответ: _____.

Рекомендации к решению заданий 22.3. В этой группе заданий, по отношению к заданиям вида 22.1, добавляется дополнительное условие про «траекторию» вычислений. Это значит, что в процессе преобразования исходного числа в конечное нужно обязательно получить определённое промежуточное число (назовём его числом T). Эти задания рекомендуется решать в два **отдельных** (!) этапа. На первом этапе — посчитать число способов получения числа T из исходного числа. На втором — посчитать число способов получения конечного числа из числа T . Результаты этапов нужно перемножить.

! В процессе такого решения следует решать именно **отдельные** задачи! То есть, ни в коем случае нельзя при вычислении второго этапа считать, что количество способов получить числа, меньшие T , уже вычислено на первом этапе! Важно на втором этапе считать, что количество способов получить из числа T числа, меньшие T , равно нулю.

22.3.1. 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 удваивает его.

Программа для исполнителя Июнь15 — это последовательность команд. Сколько существует программ, для которых при исходном числе 1 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 8, 16, 17.

Ответ: _____.

22.3.2. Исполнитель Июнь16 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 удваивает его.

Программа для исполнителя Июнь16 — это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 58, и при этом траектория вычислений содержит число 20?

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: _____.

22.3.3. 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 удваивает его.

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: _____.

22.3.4. Исполнитель Июнь16 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 удваивает его.

Программа для исполнителя Июнь16 — 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: _____.

22.3.5. 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 удваивает его.

Программа для исполнителя Июнь16 — это последовательность команд. Сколько существует программ, для которых при исходном числе **3** результатом является число **48** и при этом траектория вычислений содержит число **20**?

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: _____.

Рекомендации к решению заданий 22.4. В этой группе собраны задания, в которых, кроме ограничения «траектория содержит число Т», имеется также ограничение «траектория не содержит число Х». Для их решения нужно делать примерно то же самое, что и при решении заданий 22.3, но ещё на начальном этапе вычислений определить, что $F(X) = 0$. Например, при табличном способе записи решения рекомендуется сразу после рисования таблицы поставить в ячейку $F(X)$ число 0 (чтобы впоследствии не забыть про это). В остальном способ решения полностью совпадает с решением заданий 22.3.

22.4.1. 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 удваивает его.

Программа для исполнителя Июнь12 — это последовательность команд. Сколько существует программ, для которых при исходном числе 1 результатом является число 30, и при этом траектория вычислений содержит число 10 и не содержит число 6?

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: _____.

22.4.2. Исполнитель Увеличитель преобразует число на экране.

У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1

2. Умножить на 2

3. Умножить на 3

Первая команда увеличивает число на экране на 1, вторая удваивает его, а третья утраивает его.

Программа для исполнителя Увеличитель — это последовательность команд.

Сколько существует программ, для которых при исходном числе 2 результатом является число 45 и при этом траектория вычислений содержит число 14 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: _____.

22.4.3. Исполнитель Увеличитель преобразует число на экране.

У исполнителя есть три команды, которым присвоены номера:

1. Прибавить 1

2. Умножить на 2

3. Умножить на 3

Первая команда увеличивает число на экране на 1, вторая удваивает его, а третья утраивает его.

Программа для исполнителя Увеличитель — это последовательность команд. Сколько существует программ, для которых при исходном числе 3 результатом является число 40 и при этом траектория вычислений содержит число 13 и не содержит число 18?

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: _____.

22.4.4. 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 удваивает его, а третья утраивает его.

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: _____.

22.4.5. 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 удваивает его, а третья утраивает его.

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: _____.

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

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) = 1$$

$$(x_3 \wedge x_4) \vee (x_5 \wedge x_6) = 1$$

$$(x_5 \wedge x_6) \vee (x_7 \wedge x_8) = 1$$

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

Решение.

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:

$$\begin{cases} y_1 \vee y_2 = 1 \\ y_2 \vee y_3 = 1 \\ y_3 \vee y_4 = 1 \end{cases}$$

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 $y_1 \vee y_2 = 1$ был верным, нужно чтобы $y_2 = 1$.

Если $y_2 = 1$, то второе уравнение ($y_2 \vee y_3 = 1$) будет истинным, и мы не сможем сделать никакого вывода из этого уравнения относительно y_3 . Поэтому перейдём к рассмотрению третьего уравнения.

• Пусть $y_3 = 0$. Тогда, чтобы результат $y_3 \vee y_4 = 1$ был верным, нужно чтобы $y_4 = 1$.

Мы дошли до конца «веточки рассуждений». Выяснили, что одним из решений системы относительно переменных y является $y_1 = 0$, $y_2 = 1$, $y_3 = 0$, $y_4 = 1$. 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 $x_2 = 0$), $y_2 = (x_3 \wedge x_4) = 1$ — 1 случай (когда $x_3 = x_4 = 1$), $y_3 = 0$ — 3 случая, $y_4 = 1$ — 1 случай. Перемножаем все эти случаи: $3 \cdot 1 \cdot 3 \cdot 1 = 9$.

- Теперь пусть $y_3 = 1$. Тогда третье уравнение тоже истинно, и мы не можем сделать никакого вывода относительно y_4 . То есть, одним из решений системы относительно переменных y является $y_1 = 0$, $y_2 = 1$, $y_3 = 1$, y_4 любой. Число вариантов в этом случае относительно переменных x : $3 \cdot 1 \cdot 1 \cdot 4 = 12$.

- 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 из второго уравнения $y_3 = 1$. Тогда третье уравнение истинно, и y_4 — любое. То есть, одним из решений системы относительно переменных y является $y_1 = 1$, $y_2 = 0$, $y_3 = 1$, y_4 любой. Число вариантов в этом случае относительно переменных x : $1 \cdot 3 \cdot 1 \cdot 4 = 12$.

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

- Пусть $y_3 = 0$. Тогда $y_4 = 1$. То есть, одним из решений системы относительно переменных y является $y_1 = 1$, $y_2 = 1$, $y_3 = 0$, $y_4 = 1$. Число вариантов в этом случае относительно переменных x : $1 \cdot 1 \cdot 3 \cdot 1 = 3$.

- 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 : $1 \cdot 1 \cdot 1 \cdot 4 = 4$.

Складываем вместе все варианты решения в рассмотренных веточках рассуждений: $9 + 12 + 12 + 3 + 4 = 40$.

Ответ: 40.

❗ При решении на бумаге все приведённые слова-рассуждения писать не нужно, поэтому удобнее оформлять решение в виде дерева рассуждений.

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

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

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

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

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

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

Ответ: _____.

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

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

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

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

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

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

Ответ: _____.

23.1.3. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_9, x_{10}$, 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 $x_1, x_2, \dots, x_9, x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

23.1.4. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_9, x_{10}$, 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 $x_1, x_2, \dots, x_9, x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

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

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

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

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

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

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

Ответ: _____.

Рекомендации к решению заданий 23.2. В заданиях 23.2.1 и 23.2.2 имеет смысл сразу заметить, что вторую и третью скобки в каждой строке можно преобразовать к такому же виду, как первая скобка.

Рекомендуем решать эти задания так же, как задание 23.1, за исключением того, что в заданиях 23.2.1, 23.2.2 и 23.2.3 нельзя будет сделать замену переменных. Поэтому дерево рассуждений нужно будет строить сразу относительно переменной x_1 .

В задании 23.2.3 используется сразу два набора переменных (x и y). Имеет смысл строить дерево решений сначала по одной переменной (например, по переменной x). А потом, дойдя до конца каждой ветки рассуждений, рассматривать оставшиеся нерассмотренными варианты для y .

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

$$(x_1 \equiv x_2) \vee (x_1 \wedge x_3) \vee (\neg x_1 \wedge \neg x_3) = 1$$

$$(x_2 \equiv x_3) \vee (x_2 \wedge x_4) \vee (\neg x_2 \wedge \neg x_4) = 1$$

$$(x_3 \equiv x_4) \vee (x_3 \wedge x_5) \vee (\neg x_3 \wedge \neg x_5) = 1$$

$$(x_4 \equiv x_5) \vee (x_4 \wedge x_6) \vee (\neg x_4 \wedge \neg x_6) = 1$$

$$(x_5 \equiv x_6) \vee (x_5 \wedge x_7) \vee (\neg x_5 \wedge \neg x_7) = 1$$

$$(x_6 \equiv x_7) \vee (x_6 \wedge x_8) \vee (\neg x_6 \wedge \neg x_8) = 1$$

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

Ответ: _____.

23.2.2. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_7, x_8$, 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 $x_1, x_2, \dots, x_7, x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

23.2.3. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow (x_2 \wedge \neg y_1)) \wedge (y_1 \rightarrow y_2) = 1$$

$$(x_2 \rightarrow (x_3 \wedge \neg y_2)) \wedge (y_2 \rightarrow y_3) = 1$$

$$(x_3 \rightarrow (x_4 \wedge \neg y_3)) \wedge (y_3 \rightarrow y_4) = 1$$

$$(x_4 \rightarrow (x_5 \wedge \neg y_4)) \wedge (y_4 \rightarrow y_5) = 1$$

$$(x_5 \rightarrow (x_6 \wedge \neg y_5)) \wedge (y_5 \rightarrow y_6) = 1$$

$$(x_6 \rightarrow (x_7 \wedge \neg y_6)) \wedge (y_6 \rightarrow y_7) = 1$$

$$x_7 \rightarrow \neg y_7 = 1$$

В ответе **не нужно** перечислять все различные наборы значений $x_1, x_2, \dots, x_7, y_1, y_2, \dots, y_7$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

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

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

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

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

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

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

Ответ: _____.

23.2.5. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_9, x_{10}$, 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 $x_1, x_2, \dots, x_9, x_{10}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

Рекомендации к решению заданий 23.3. В заданиях 23.3.1 и 23.3.2 нельзя будет сделать замену переменных. Но можно заметить, что переменные одной пары (например, x_1 и x_2) взаимодействуют с переменными другой пары только в парах. Поэтому имеет смысл при построении дерева рассуждений рассматривать сразу пары переменных.

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

$$(x_1 \rightarrow x_2) \wedge ((x_1 \equiv x_2) \rightarrow (x_3 \equiv x_4)) = 1$$

$$(x_3 \rightarrow x_4) \wedge ((x_3 \equiv x_4) \rightarrow (x_5 \equiv x_6)) = 1$$

$$(x_5 \rightarrow x_6) \wedge ((x_5 \equiv x_6) \rightarrow (x_7 \equiv x_8)) = 1$$

$$(x_7 \rightarrow x_8) \wedge ((x_7 \equiv x_8) \rightarrow (x_9 \equiv x_{10})) = 1$$

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

Ответ: _____.

23.3.2. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_{11}, x_{12}$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \rightarrow x_2) \wedge (x_2 \rightarrow x_1) \wedge (x_3 \rightarrow x_4) = 1$$

$$(x_3 \rightarrow x_4) \wedge (x_4 \rightarrow x_3) \wedge (x_5 \rightarrow x_6) = 1$$

$$(x_5 \rightarrow x_6) \wedge (x_6 \rightarrow x_5) \wedge (x_7 \rightarrow x_8) = 1$$

$$(x_7 \rightarrow x_8) \wedge (x_8 \rightarrow x_7) \wedge (x_9 \rightarrow x_{10}) = 1$$

$$(x_9 \rightarrow x_{10}) \wedge (x_{10} \rightarrow x_9) \wedge (x_{11} \rightarrow x_{12}) = 1$$

В ответе не нужно перечислять все различные наборы значений $x_1, x_2, \dots, x_{11}, x_{12}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

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

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

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

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

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

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

Ответ: _____.

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

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

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

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

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

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

Ответ: _____.

В задании **23.3.5** имеет смысл сначала изучить левый столбец условий. И сделать вывод, что в каждой паре переменных не может быть одновременно обоих нулей. После этого (помня постоянно об этом ограничении) можно решать задание правого столбца, не обращая внимания на левый столбец, в котором можно рассматривать переменные парами (или даже сделать замену переменных).

23.3.5. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_{11}, x_{12}$, которые удовлетворяют всем перечисленным ниже условиям?

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

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

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

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

$$(x_9 \vee x_{10}) \wedge ((\neg x_9 \vee \neg x_{10}) \rightarrow (\neg x_{11} \vee \neg x_{12})) = 1$$

$$(x_{11} \vee x_{12}) = 1$$

В ответе **не нужно** перечислять все различные наборы значений $x_1, x_2, \dots, x_{11}, x_{12}$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

Рекомендации к решению заданий 23.4. К решению заданий этой группы нет особенных рекомендаций, отличных от тех, что уже были приведены раньше. Эта группа приведена здесь для того, чтобы вы закрепили навыки решения подобных задач. Не забывайте перед решением задачи изучить исходное выражение на тему «нельзя ли свернуть длинное выражение в виде короткой операции».

Стройте дерево решений и не «теряйте» варианты ответов.

23.4.1. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_6, x_7$, которые удовлетворяют всем перечисленным ниже условиям?

$$(x_1 \equiv \neg x_2) \vee (x_1 \wedge \neg x_3) \vee (\neg x_1 \wedge x_3) = 1$$

$$(x_2 \equiv \neg x_3) \vee (x_2 \wedge \neg x_4) \vee (\neg x_2 \wedge x_4) = 1$$

$$(x_3 \equiv \neg x_4) \vee (x_3 \wedge \neg x_5) \vee (\neg x_3 \wedge x_5) = 1$$

$$(x_4 \equiv \neg x_5) \vee (x_4 \wedge \neg x_6) \vee (\neg x_4 \wedge x_6) = 1$$

$$(x_5 \equiv \neg x_6) \vee (x_5 \wedge \neg x_7) \vee (\neg x_5 \wedge x_7) = 1$$

В ответе не нужно перечислять все различные наборы значений $x_1, x_2, \dots, x_6, x_7$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

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

$$(x_1 \rightarrow x_2) \vee (x_1 \rightarrow x_3) = 1$$

$$(x_2 \rightarrow x_3) \vee (x_2 \rightarrow x_4) = 1$$

$$(x_3 \rightarrow x_4) \vee (x_3 \rightarrow x_5) = 1$$

$$(x_4 \rightarrow x_5) \vee (x_4 \rightarrow x_6) = 1$$

$$(x_5 \rightarrow x_6) \vee (x_5 \rightarrow x_7) = 1$$

$$(x_6 \rightarrow x_7) \vee (x_6 \rightarrow x_8) = 1$$

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

Ответ: _____.

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

$$(x_1 \wedge x_2) \vee (x_2 \wedge x_3) = 0$$

$$(x_2 \wedge x_3) \vee (x_3 \wedge x_4) = 0$$

$$(x_3 \wedge x_4) \vee (x_4 \wedge x_5) = 0$$

$$(x_4 \wedge x_5) \vee (x_5 \wedge x_6) = 0$$

$$(x_5 \wedge x_6) \vee (x_6 \wedge x_7) = 0$$

$$(x_6 \wedge x_7) \vee (x_7 \wedge x_8) = 0$$

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

Ответ: _____.

23.4.4. Сколько существует различных наборов значений логических переменных $x_1, x_2, \dots, x_7, x_8$, 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 $x_1, x_2, \dots, x_7, x_8$, при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

Ответ: _____.

23.4.5. 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?

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

$$(y_1 \rightarrow y_2) \wedge (y_2 \rightarrow y_3) \wedge (y_3 \rightarrow y_4) \wedge (y_4 \rightarrow y_5) = 1$$

$$(\neg y_1 \vee x_1) \wedge (\neg y_2 \vee x_2) \wedge (\neg y_3 \vee x_3) \wedge (\neg y_4 \vee x_4) \wedge (\neg y_5 \vee x_5) = 1$$

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: _____.

24

АНАЛИЗ ПРОГРАММЫ. ИСПРАВЛЕНИЕ ОШИБОК В ПРОГРАММЕ

24.1. Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран наименьшую нечётную цифру числа. Если в числе нет чётных цифр, программа должна вывести на экран «NO». Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre> var N, d, m: integer; begin readln(N); m := 9; while N > 0 do begin d := N mod 10; if d < m then if d mod 2 <> 0 then m := d; N := N div 10 end; if m = 9 then writeln('NO') else writeln(m) end. </pre>	<pre> DIM N AS LONG INPUT N m = 9 WHILE N > 0 d = N MOD 10 IF d < m THEN IF d MOD 2 <> 0 THEN m = d END IF END IF N = N \ 10 WEND IF m = 9 THEN PRINT "NO" ELSE PRINT m END IF END </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { long int N; int d, m; cin >> N; m = 9; while (N > 0) { d = N % 10; if (d < m) if (d % 2 != 0) m = d; N = N / 10; } if (m == 9) printf("NO"); else cout << m << endl; } </pre>	<pre> алг нач цел N, m, d ввод N m := 9 нц пока N > 0 d := mod(N, 10) если d < m то если mod(d, 2) <> 0 то m := d все все N := div(N, 10) кц если m = 9 то вывод 'NO' иначе вывод m все кон </pre>

Python

```
N = int(input())
m = 9
while N > 0:
    d = N % 10
    if d < m:
        if d % 2 != 0:
            m = d
    N = N // 10
if m == 9:
    print("NO")
else:
    print(m)
```

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 496.
2. Приведите такое наибольшее трёхзначное число, 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 496.

Рассмотрим подробную сортировку данной программы на языке Паскаль.

Оператор	Условие	Переменные			Пояснения
		N	d	m	
readln(N);		496			Считываем с клавиатуры значение N
m:=9;				9	Переменная m получает начальное значение (9)
while N>0 do	496>0? Да				Проверяем условие цикла. Оно выполняется
d:=N mod 10;			6		Переменная d получает значение младшего разряда переменной N
if d<m then	6<9? Да				Переменная d (6) меньше m (9)? Да. Переходим к проверке в следующей строке.

Окончание табл.					
Оператор	Условие	Переменные			Пояснения
		N	d	m	
if d mod 2<>0 then	0<>0? Нет				Переменная d — нечётная? Нет. Пропускаем следующую строку
N:=N div 10		49			Из переменной N удаляем младший разряд
while N>0 do	49>0? Да				Проверяем условие цикла. Оно выполняется
d:=N mod 10;			9		Переменная d получает значение младшего разряда переменной N
if d < m then	9<9? Да				Переменная d (9) меньше m (9)? Нет. Пропускаем следующие две строки
N:=N div 10		4			Из переменной N удаляем младший разряд
while N>0 do	4>0? Да				Проверяем условие цикла. Оно выполняется
d:=N mod 10;			4		Переменная d получает значение младшего разряда переменной N
if d<m then	4<9? Да				Переменная d (4) меньше m (9)? Да
if d mod 2<>0 then	0<>0? Нет				Переменная d — нечётная? Нет. Пропускаем следующую строку
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 m=9 then	9=9? Да				Проверяем условие вывода ответа
writeln('NO')					На экране: NO
На чистовике пишем в ответ на вопрос 1: NO Понимаем, что программа выдала «нет нечётных цифр» несмотря на то, что во вводимом числе N есть нечётная цифра — 9.					

Анализируем программу и понимаем, что последнее условие ($m=9$) некорректно проверяет наличие нечётной цифры, если в числе нет нечётных цифр, кроме 9. Значит, эта проверка должна быть какой-то другой. Однако это условие проверяет лишь то, что переменная m не изменилась и осталась равна исходному значению (9). Следовательно, ошибка именно в исходном значении переменной m . Действительно, хотя число 9 является самым большим значением цифры, принимать его в качестве начального значения минимума неверно. Таким способом мы не сможем после отличить случай, когда в числе нет нечётных цифр от случая, когда в числе есть нечётные цифры, но эти цифры — 9. Значит, в качестве начального значения минимума (переменной m) нужно взять число, которое не может являться верным результатом поиска. Например, для этого подойдёт число 10. Таким образом, следует задать начальное значение $m=10$, а в конце программы проверить, не осталось ли $m=10$. Получается, мы нашли две ошибки в программе:

Неверно	Верно
$m:=9;$	$m:=10;$
if $m=9$ then	if $m=10$ then

Проверим ещё раз получившуюся программу. Для этого выполним трассировку исправленной программы для числа из первого вопроса — 496. Убеждаемся, что программа выводит правильный ответ. Хорошо бы ещё проверить программу для других случаев, например, для числа 486. В нём нет нечётных цифр, и программа должна выдать ответ «NO». Также для числа 235. В нём несколько нечётных цифр. Программа должна выдать наименьшую нечётную цифру — 3. 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 неверно работает при наличии в числе нечётной цифры 9 и отсутствию нечётных цифр, меньших 9. То есть, программа, возможно, правильно работает при наличии в числе других нечётных цифр или при отсутствии нечётных цифр. Требуется подобрать наибольшее трёхзначное число. Подберём такое в каждой категории. Наибольшее трёхзначное число, в котором есть нечётные цифры, меньшие 9: 997. Наибольшее трёхзначное число, в котором нет нечётных цифр: 888. Подставим 997 в исходную программу и проверим, выдаёт ли она при этом правильный ответ (7). Выполним трассировку (можно сделать подробную трассировку, как при ответе на вопрос 1, или сделать упрощённую трассировку).

Важно при этом не принимать желаемое за действительное, т. е. 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 7. Запишем в качестве ответа на вопрос 2 число 997.

Ответ:

1. NO
2. 997

3.

Неверно	Верно
m:=9;	m:=10;
if m=9 then	if m=10 then

❗ Рекомендуем ни в коем случае не записывать в качестве ответа ничего кроме указанного. 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!

24.1.1. Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран наибольшую чётную цифру числа. Если в числе нет чётных цифр, программа должна вывести на экран «NO». Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre> var N, d, m: integer; begin readln(N); m := N mod 10; while N > 0 do begin d := N mod 10; if d > m then if d mod 2 = 0 then m := d; N := N div 10 end; if m = 0 then writeln('NO') else writeln(m) end. </pre>	<pre> DIM N AS LONG INPUT N m = N MOD 10 WHILE N > 0 d = N MOD 10 IF d > m THEN IF d MOD 2 = 0 THEN m = d END IF END IF N = N \ 10 WEND IF m = 0 THEN PRINT "NO" ELSE PRINT m END IF END </pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { long int N; int d, m; cin >> N; m = N % 10; while (N > 0) { d = N % 10; if (d > m) if (d % 2 == 0) m = d; N = N / 10; } if (m == 0) printf("NO"); else cout << m << endl; }</pre>	<pre>алг нач цел N, m, d ввод N m := mod(N, 10) нц пока N > 0 d := mod(N, 10) если d > m то если mod(d, 2) = 0 то m := d все N := div(N, 10) кц если m = 0 то вывод 'NO' иначе вывод m все кон</pre>
Python	
<pre>N = int(input()) m = N % 10 while N > 0: d = N % 10 if d > m: if d % 2 == 0: m = d N = N // 10 if m == 0: print("NO") else: print(m)</pre>	

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.

24.1.2. Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран наибольшую цифру числа, которая меньше 6. Если в числе нет цифр, меньших 6, программа должна вывести на экран «NO». Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre> var N, d, m: integer; begin readln(N); m := 0; while N > 0 do begin d := N mod 10; if d < 6 then if d > m then m := d; N := N div 10 end; if m > 0 then writeln(m) else writeln('NO') end. </pre>	<pre> DIM N AS LONG INPUT N m = 0 WHILE N > 0 d = N MOD 10 IF d < 6 THEN IF d > m THEN m = d END IF END IF N = N \ 10 WEND IF m > 0 THEN PRINT m ELSE PRINT "NO" END IF END </pre>
C++	Алгоритмический язык
<pre> #include <iostream> using namespace std; int main() { long int N; int d, m; cin >> N; m = 0; while (N > 0) { d = N % 10; if (d < 6) if (d > m) m = d; N = N / 10; } if (m > 0) cout << m << endl; else cout << "NO" << endl; } </pre>	<pre> алг нач цел N, m, d ввод N m := 0 нц пока N > 0 d := mod(N, 10) если d < 6 то если d > m то m := d все все N := div(N, 10) кц если m > 0 то вывод m иначе вывод 'NO' все кон </pre>

Python
<pre> N = int(input()) m = 0 while N > 0: d = N % 10 if d < 6: if d > m: m = d N = N // 10 if m > 0: print(m) else: print("NO") </pre>

Выполните следующие действия:

1. Напишите, что выведет программа при вводе числа 706.
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.

24.1.3. 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	Бейсик
<pre> 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. </pre>	<pre> 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 </pre>

C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N, sum, d <u>ввод</u> N sum := 0 <u>нц</u> <u>пока</u> N > 1 d := mod(N,10) <u>если</u> mod(d,2) <> 0 <u>то</u> sum := sum + d <u>все</u> N := div(N, 10) <u>кц</u> <u>вывод</u> d <u>кон</u></pre>
Python	
<pre>N = int(input()) sum = 0 while N > 1: d = N % 10 if d % 2 != 0: sum = sum + d N = N // 10 print(d)</pre>	

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.

24.1.4. 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	Бейсик
<pre>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.</pre>	<pre>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</pre>
C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N, k <u>ввод</u> N k := 10 <u>нц пока</u> N > 0 <u>если</u> mod(N,10) < k <u>то</u> k := mod(N,10) <u>все</u> N := div(N, 10) <u>кц</u> <u>вывод</u> k <u>кон</u></pre>
Python	
<pre>N = int(input()) k = 10 while N > 0: if N % 10 < k: k = N % 10 N = N // 10 print(k)</pre>	

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.

24.1.5. Требовалось написать программу, которая вводит с клавиатуры натуральное число N ($N < 10^9$) и выводит на экран сумму цифр числа. Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre>var N, s: integer; begin readln(N); s := 0; while N > 0 do begin N := N div 10; s := s + N mod 10 end; writeln(s) end.</pre>	<pre>DIM N AS LONG INPUT N s = 0 WHILE N > 0 N = N \ 10 s = s + N MOD 10 WEND PRINT s END</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { long int N; int s; cin >> N; s = 0; while (N > 0) { N = N / 10; s = s + N % 10; } cout << s; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> N, s <u>ввод</u> N s := 0 <u>нц пока</u> N > 0 N := div(N, 10) s := s + mod(N, 10) <u>кц</u> <u>вывод</u> s <u>кон</u></pre>
Python	
<pre>N = int(input()) s = 0 while N > 0: N = N // 10 s = s + N % 10 print(s)</pre>	

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.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

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

Паскаль	Бейсик
<pre>const n = 4; var i, a, max1: integer; begin max1 := -1; for i := 1 to n do begin read(a); if a mod 2 = 0 then if a > max1 then max1 := a end; if max1 <> -1 then writeln(max1) else writeln('NO') end. end;</pre>	<pre>CONST N = 4 DIM I,A,MAX1 AS INTEGER MAX1 := -1 FOR I = 1 TO N INPUT A IF A MOD 2 = 0 THEN IF A > MAX1 THEN MAX1 = A END IF END IF NEXT I IF MAX1 <> -1 THEN PRINT MAX1 ELSE PRINT "NO" END IF END</pre>

Python

```

N = 4
max1 = -1
for i in range(n):
    a = int(input())
    if a % 2 == 0:
        if a > max1:
            max1 = a
if max1 != -1:
    print(max1)
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.

Решение.

Выполним трассировку программы для данной последовательности 1 —2 3 —4.

Рассмотрим подробную сортировку данной программы на языке Паскаль.

Оператор	Условие	Переменные			Пояснения
		i	a	max1	
max1 := -1;				-1	Переменная max1 получает начальное значение (-1)
for i:=1 to n do	1≤4? Да	1			Начинается цикл for. Переменная i получает начальное значение (1)
read(a);			1		Считываем значение элемента последовательности
if a mod 2=0 then	1=0? Нет				Проверяем переменную a (1) на чётность. Нет, не чётная. Пропускаем следующие две строки

Продолжение табл.

Оператор	Условие	Переменные			Пояснения
		i	a	max1	
for i:=1 to n do	$2 \leq 4?$ Да	2			Переменная i получает следующее значение (2). Цикл for продолжается.
read(a);			-2		Считываем значение элемента последовательности
if a mod 2=0 then	$0=0?$ Да				Проверяем переменную a (-2) на чётность. Да, чётная. Переходим к следующей строке
if a>max1 then	$-2 > -1?$ Нет				Проверяем условие максимума. Не выполнилось. Пропускаем следующую строку
for i:=1 to n do	$3 \leq 4?$ Да	3			Переменная i получает следующее значение (3). Цикл for продолжается.
read(a);			3		Считываем значение элемента последовательности
if a mod 2=0 then	$1=0?$ Нет				Проверяем переменную a (3) на чётность. Нет, не чётная. Пропускаем следующие две строки
for i:=1 to n do	$4 \leq 4?$ Да	4			Переменная i получает следующее значение (4). Цикл for продолжается.
read(a);			-4		Считываем значение элемента последовательности
if a mod 2=0 then	$0=0?$ Да				Проверяем переменную a (-4) на чётность. Да, чётная. Переходим к следующей строке

Окончание табл.

Оператор	Условие	Переменные			Пояснения
		i	a	max1	
if a>max1 then	$-4 > -1$? Нет				Проверяем условие максимума. Не выполнилось. Пропускаем следующую строку
for i:=1 to n do	$5 \leq 4$? Да	5			Переменная i получает следующее значение (5). Цикл for заканчивается.
if max1<>-1 then	$-1 \neq -1$? Нет				Проверяем условие вывода ответа. Не выполняется
writeln('NO')					На экран выводим «NO»

В качестве ответа на первый вопрос пишем: NO

Несмотря на наличие во входной последовательности чётных чисел, программа вывела на экран «NO». Сначала проверим, верно ли осуществляется проверка на чётность (проверяется остаток от деления на 2 на равенство нулю). Да, верно. Тогда заметим по трассировке, что при нахождении очередного чётного элемента последовательности программа не меняла переменную max1, потому что она оказывалась больше текущего значения чётного элемента последовательности. Значит, начальное значение переменной max1 (−1), хотя и является нечётным, не позволяет верно найти наибольший чётный элемент, если он будет отрицательным. Следовательно, в качестве начального значения переменной max1 нужно выбрать такое значение, которое будет заведомо меньше любого чётного отрицательного элемента последовательности среди возможных. По условию значения элементов последовательности не превышают по модулю 1000. Значит, нужно взять начальное значение max1 = −1001. При проверке, какой ответ выводить на экран, нужно проверять, что max1 не равен −1001.

Получаем:

Неверно	Верно
max1:=-1;	max1:=-1001;
if max1<>-1 then	if max1<>-1001 then

Проверим получившуюся программу.

Выполним трассировку исправленной программы хотя бы для данной последовательности 1 —2 3 —4.

Хорошо бы также проверить работу программы для других случаев. Например, для случая, когда чётных элементов нет (1 3 5 7). И для случая, когда чётные элементы и положительные и отрицательные (1 —2 4 —5). После проверки запишем эту таблицу в качестве ответа на вопрос 3.

Найдём ответ на вопрос 2.

При анализе программы для поиска ошибок можно было заметить, что ошибка происходит при отрицательных чётных элементах последовательности. Попробуем взять последовательность, в которой чётные элементы положительны: 1 2 3 4.

Выполним трассировку исходной программы для этой последовательности (оставляем трассировку на вас). Убеждаемся, что программа верно выдаёт значение наибольшего чётного числа в последовательности (4). Запишем эту последовательность в качестве ответа на вопрос 2.

Получаем окончательный ответ:

1. NO
2. 1 2 3 4
- 3.

Неверно	Верно
max1:=-1;	max1:=-1001;
if max1<>-1 then	if max1<>-1001 then

24.2.1. 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	Бейсик
<pre> 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. </pre>	<pre> 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 </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 </pre>

Python
<pre> 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) </pre>

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.

24.2.2. Требовалось написать программу, которая вводит с клавиатуры последовательность из n целых чисел ($n = 4$), и выводит на экран количество нечётных чисел среди этой последовательности.

Известно, что вводимые числа не превышают по модулю 1000. Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre> const n = 4; var i, a, count: integer; begin count := 0; for i := 1 to n do begin read(a); if a mod 2 > 0 then count := count + a end; writeln(count) end. </pre>	<pre> CONST N = 4 DIM I, A, COUNT AS INTEGER COUNT := 0 FOR I = 1 TO N INPUT A IF A MOD 2 > 0 THEN COUNT = COUNT + A END IF NEXT I PRINT COUNT END </pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; const int n = 4; int main() { int i, a, count; count = 0; for (i=0 ; i<n ; i++) { cin >> a; if(a % 2 > 0) count = count + a; } cout << count << endl; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> n = 4 <u>цел</u> a, i, count count := 0 <u>нц для i от 1 до n</u> <u>ввод</u> a <u>если</u> mod(a,2) > 0 <u>то</u> count := count + a <u>все</u> <u>кц</u> <u>вывод</u> count <u>кон</u></pre>
Python	
<pre>n = 4 count = 0 for i in range(n): a = int(input()) if a % 2 > 0: count = count + a print(count)</pre>	

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел -1 1 -3 5 .
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.

24.2.3. 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	Бейсик
<pre>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. end.</pre>	<pre>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</pre>
C++	Алгоритмический язык
<pre>#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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> n = 4 <u>цел</u> a, i, max max := 0 <u>нц для</u> i <u>от</u> 1 <u>до</u> n <u>ввод</u> a <u>если</u> a > max <u>то</u> a := max <u>все</u> <u>кц</u> <u>вывод</u> max <u>кон</u></pre>
Python	
<pre>n = 4 max = 0 for i in range(n): a = int(input()) if a > max: a = max print(max)</pre>	

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.

24.2.4. 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	Бейсик
<pre>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. end.</pre>	<pre>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</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; const int 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; }</pre>	<pre><u>алг</u> <u>нач</u> <u>цел</u> n = 4 <u>цел</u> a, i, sum, count sum := 0 count := 1 <u>нц для</u> i <u>от</u> 1 <u>до</u> n <u>ввод</u> a <u>если</u> mod(a,2) = 0 <u>то</u> sum := sum + a count := count + 1 <u>все</u> <u>кц</u> <u>если</u> sum > 0 <u>то</u> <u>вывод</u> sum <u>иначе</u> <u>вывод</u> 'NO' <u>все</u> <u>кон</u></pre>

Python
<pre> 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") </pre>

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.

24.2.5. Требовалось написать программу, которая вводит с клавиатуры последовательность из n целых чисел ($n = 4$), и выводит на экран сумму отрицательных чисел среди этой последовательности. Если в последовательности нет отрицательных чисел, программа должна вывести 0. Известно, что вводимые числа не превышают по модулю 1000. Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre> const n = 4; var i, a, sum: integer; begin sum := 0; for i := 1 to n do begin read(a); if a < 0 then sum := a end; writeln(a) end. </pre>	<pre> CONST N = 4 DIM I, A, SUM AS INTEGER SUM := 0 FOR I = 1 TO N INPUT A IF A < 0 THEN SUM = A END IF NEXT I PRINT A END </pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; const int n = 4; int main() { int i, a, sum; sum = 0; for (i=0 ; i<n ; i++) { cin >> a; if(a < 0) sum = a; } cout << a << endl; }</pre>	<pre>алг нач цел n = 4 цел a, i, sum sum := 0 нц для i от 1 до n ввод a если a < 0 то sum := a все кц вывод a кон</pre>
Python	
<pre>n = 4 sum = 0 for i in range(n): a = int(input()) if a < 0: sum = a print(a)</pre>	

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел $-1\ 1\ 3\ -5$.
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.

24.2.6. Требовалось написать программу, которая вводит с клавиатуры последовательность целых ненулевых чисел, оканчивающихся 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	Бейсик
<pre> 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. </pre>	<pre> 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 </pre>
C++	Алгоритмический язык
<pre> #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; } </pre>	<pre> 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 mod(a,2) = 0 то вывод 'NO' иначе вывод s / c все кон </pre>

Python
<pre> 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) </pre>

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.

24.2.7. Требовалось написать программу, которая вводит с клавиатуры последовательность из n натуральных чисел ($n = 4$), и выводит на экран количество чисел среди этой последовательности, которые оканчиваются на 5. Программист торопился и написал программу неправильно.

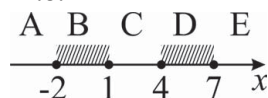
Паскаль	Бейсик
<pre> const n = 4; var i, a, s: integer; begin s := 0; for i := 1 to n do begin read(a); if a mod 5 = 0 then s := s + a end; writeln(s) end. </pre>	<pre> CONST N = 4 DIM I, A, MIN AS INTEGER s = 0 FOR I = 1 TO N INPUT a IF a MOD 5 = 0 THEN s = s + a ENDIF NEXT I PRINT s END </pre>

C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; const int n = 4; int main() { int i, a, s; s = 0; for (i=0 ; i<n ; i++) { cin >> a; if(a % 5 == 0) s = s + a; } cout << s << endl; }</pre>	<pre>алг нач цел n = 4 цел i, a, s s := 0 нц для i от 1 до n ввод a если mod(a,5) = 0 то s := s + a все кц вывод s кон</pre>
Python	
<pre>n = 4 s = 0 for i in range(n): a = int(input()) if a % 5 == 0: s = s + a print(s)</pre>	

Выполните следующие действия:

1. Напишите, что выведет программа при вводе чисел 1 5 10 15.
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.

24.3.



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 В и D (включая границы). Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre>var x: real; begin readln(x); if x >= -2 then if x <= 7 then if x < 4 then write('не принадлежит') else write('принадлежит') end. end.</pre>	<pre>INPUT x IF x >= -2 THEN IF x <= 7 THEN IF x < 4 THEN PRINT "не принадлежит" ELSE PRINT "принадлежит" ENDIF ENDIF ENDIF END</pre>
Си	Алгоритмический язык
<pre>void main(void) { float x; scanf("%f",&x); if (x >= -2) if (x <= 7) if (x < 4) printf("не принадлежит"); else printf("принадлежит"); } }</pre>	<pre><u>алг</u> <u>нач</u> <u>вещ</u> x, y <u>ввод</u> x, y <u>если</u> x >= -2 <u>то</u> <u>если</u> x <= 7 <u>то</u> <u>если</u> x < 4 <u>то</u> <u>вывод</u> 'не принадлежит' <u>иначе</u> <u>вывод</u> 'принадлежит' <u>все</u> <u>все</u> <u>все</u> <u>кон</u></pre>

Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D и Е). Границы (точки -2 , 1 , 4 и 7) принадлежат заштрихованным областям (В и D соответственно).

Область	Условие 1 ($x \geq -2$)	Условие 2 ($x \leq 7$)	Условие 3 ($x < 4$)	Программа выведет	Область обрабатыва- ется верно
А					
В					
С					
D					
Е					

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.)

Решение.

Заполним приведённую таблицу. Будем заполнять таблицу по столбцам. Так как сначала в программе проверяется Условие 1, запишем результат проверки этого условия для каждой указанной области (от А до Е).

Область	Условие 1 ($x \geq -2$)	Условие 2 ($x \leq 7$)	Условие 3 ($x < 4$)	Программа выведет	Область обрабатыва- ется верно
А	Нет				
В	Да				
С	Да				
Д	Да				
Е	Да				

На следующем этапе очень важно заметить, что таблицу нужно заполнять на основании работы программы, а не независимо от нее. Так, в частности, нужно чётко понимать, что Условие 2 в программе «подчинено» Условию 1. То есть, оно проверяется только в том случае, если Условие 1 верно.

А что делает программа, если Условие 1 неверно? На первый взгляд может показаться, что программа перейдёт на строчку `else` и выведет на экран «не принадлежит». Однако это совершенно не так. Нужно вспомнить, что оператор `else` относится к последнему оператору `if`, у которого ещё нет своего `else` (если не используются операторные скобки). В данном случае `else` относится только к третьему оператору `if`. То есть, у первого оператора `if` (Условие 1 которого мы только что проверили) нет своего `else`.

Значит, при невыполнении Условия 1 программа сразу заканчивает свою работу. То есть, в строке для области А правее слова «Нет» нужно будет поставить три прочерка (Условие 2 и Условие 3 не проверяются, программа ничего не выведет на экран). В последнем столбце можно сразу написать «Нет», потому что программа должна была вывести «не принадлежит», а не выводит ничего. Теперь для областей В-Е запишем результат проверки Условия 2. Текущая таблица:

Область	Условие 1 ($x \geq -2$)	Условие 2 ($x \leq 7$)	Условие 3 ($x < 4$)	Программа выведет	Область обрабатывается верно
А	Нет	—	—	—	Нет
В	Да	Да			
С	Да	Да			
Д	Да	Да			
Е	Да	Нет			

Аналогично проверке Условия 1, если Условие 2 не выполнилось, программа сразу завершается. В строке для области Е пишем прочерки в столбце Условия 3 (оно не будет проверяться), «Программа выведет» (программа ничего не выведет) и «Нет» в последнем столбце (программа должна была вывести «не принадлежит», а не вывела ничего). Для оставшихся строк (В, С, Д) запишем результат проверки Условия 3. Если Условие 3 выполнилось, программа выводит «не принадлежит». А если не выполнилось, то выводит «принадлежит». Сопоставляем результат вывода программы с тем, что она должна была вывести. Замечаем, что для областей С и Д программа выводит верный ответ. А для области В — неверный. Отметим это словами «Да» и «Нет» в последнем столбце.

Итоговая таблица:

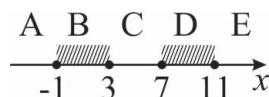
Область	Условие 1 ($x \geq -2$)	Условие 2 ($x \leq 7$)	Условие 3 ($x < 4$)	Программа выведет	Область обрабатывается верно
А	Нет	—	—	—	Нет
В	Да	Да	Да	не принадлежит	Нет
С	Да	Да	Да	не принадлежит	Да
Д	Да	Да	Нет	принадлежит	Да
Е	Да	Нет	—	—	Нет

Переходим к выполнению пункта 2. Понимаем, что в программе две ошибки. Во-первых, не проверяется расположение значения x относительно числа 1.

В результате этого для области В программа выдаёт неверный ответ. Во-вторых, в случае невыполнения Условия 1 или Условия 2 программа вообще ничего не выводит на экран, потому что у этих условий не обрабатывается случай `else`. Соответственно, можно добавить в программу ещё один `if` и ещё три `else`, либо переписать программу с использованием логических операций. А именно:

```
if (x>=-2)and(x<=1) or (x>=4)and(x<=7) then
    writeln('принадлежит')
else
    writeln('не принадлежит')
```

24.3.1. 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 В и D (включая границы). Программист торопился и написал программу неправильно.



Паскаль	Бейсик
<pre>var x: real; begin readln(x); if x <= 11 then if x >= -1 then if x < 7 then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x IF x <= 11 THEN IF x >= -1 THEN IF x < 7 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { float x; cin >> x; if (x <= 11) if (x >= -1) if (x < 7) cout<<"принадлежит"<<endl; else cout<<"не принадлежит"<<endl; }</pre>	<pre>алг нач вещ x ввод x если x <= 11 то если x >= -1 то если x < 7 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все все кон</pre>

Python
<pre> x = float(input()) if x <= 11: if x >= -1: if x < 7: print("принадлежит") else: print("не принадлежит") </pre>

Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D и Е). Границы (точки -1 , 3 , 7 и 11) принадлежат заштрихованным областям (В и D соответственно).

Область	Условие 1 ($x \leq 11$)	Условие 2 ($x \geq -1$)	Условие 3 ($x < 7$)	Программа выведет	Область обрабатыва- ется верно
А					
В					
С					
D					
Е					

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.)

24.3.2. 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 В и D (включая границы). Программист торопился и написал программу неправильно.

Паскаль	Бейсик
<pre>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. end.</pre>	<pre>INPUT x IF x > 1 THEN IF x >= 7 THEN IF x <= 13 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>
C++	Алгоритмический язык
<pre>#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"; } }</pre>	<pre><u>алг</u> <u>нач</u> <u>вещ</u> x <u>ввод</u> x <u>если</u> x > 1 <u>то</u> <u>если</u> x >= 7 <u>то</u> <u>вывод</u> 'принадлежит' <u>иначе</u> <u>вывод</u> 'не принадлежит' <u>все</u> <u>все</u> <u>все</u> <u>кон</u></pre>
Python	
<pre>x = float(input()) if x > 1: if x >= 7: if x <= 13: print("принадлежит") else: print("не принадлежит") else: print("не принадлежит")</pre>	

Последовательно выполните следующее:

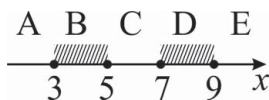
1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D и Е). Границы (точки -5, 1, 7 и 13) принадлежат заштрихованным областям (В и D соответственно).

Область	Условие 1 ($x > 1$)	Условие 2 ($x \geq 7$)	Условие 3 ($x \leq 13$)	Программа выведет	Область обрабатыва- ется 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 ее неправильной работы. (Это можно сделать несколькими способами, поэтому можно указать любой способ доработки исходной программы.)

24.3.3. 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 В и D (включая границы). Программист торопился и написал программу неправильно.



Паскаль	Бейсик
<pre> 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 end end.</pre>	<pre> INPUT x IF x >= 3 THEN IF x <= 9 THEN IF x >= 7 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF END</pre>

C++	Алгоритмический язык
<pre>#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"; } } }</pre>	<pre>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</pre>
Python	
<pre>x = float(input()) if x >= 3: if x <= 9: if x >= 7: print("принадлежит") else: print("не принадлежит") else: print("не принадлежит")</pre>	

Последовательно выполните следующее:

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D и Е). Границы (точки 3, 5, 7 и 9) принадлежат заштрихованным областям (В и D соответственно).

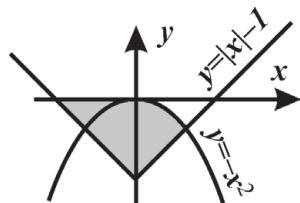
Область	Условие 1 ($x \geq 3$)	Условие 2 ($x \leq 9$)	Условие 3 ($x \geq 7$)	Программа выведет	Область обрабатыва- ется верно
А					
В					
С					
D					
Е					

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, напиши-

те «—» (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, поэтому можно указать любой способ доработки исходной программы.)

24.3.4. 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.



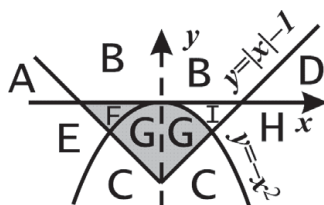
Паскаль	Бейсик
<pre>var x, y: real; begin readln(x,y); if y >= abs(x) - 1 then if y <= 0 then if y <= -x * x then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x, y IF y >= abs(x) - 1 THEN IF y <= 0 THEN IF y <= -x * x THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>
C++	Алгоритмический язык
<pre>#include <iostream> #include <cstdlib> using namespace std; int main() { float x, y; cin >> x >> y; if (y >= fabs(x) - 1) if (y <= 0) if (y <= -x * x) cout << "принадлежит"; else cout << "не принадлежит"; }</pre>	<pre><u>алг</u> <u>нач</u> <u>вещ</u> x, y <u>ввод</u> x, y <u>если</u> y >= abs(x) - 1 <u>то</u> <u>если</u> y <= 0 <u>то</u> <u>вывод</u> 'принадлежит' <u>иначе</u> <u>вывод</u> 'не принадлежит' <u>все</u> <u>все</u> <u>все</u> <u>кон</u></pre>

Python
<pre> x = float(input()) y = float(input()) if y >= abs(x) - 1: if y <= 0: if y <= -x * x: print("принадлежит") else: print("не принадлежит") </pre>

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.

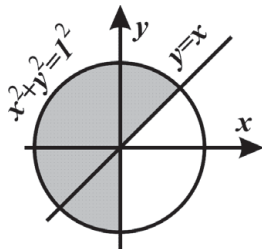


Область	Условие 1 ($y \geq \text{abs}(x) - 1$)	Условие 2 ($y \leq 0$)	Условие 3 ($y \leq -x * x$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					
I					

В столбцах условий укажите «да», если условие выполнится, «нет» если условие не выполнится, «—» (прочерк), если условие не будет проверяться, «не изв.», если программа 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, достаточно указать любой способ доработки исходной программы.)

24.3.5. 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.

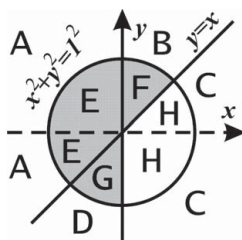


Паскаль	Бейсик
<pre>var x, y: real; begin readln(x, y); if x * x + y * y <= 1 then if y >= x then if x <= 0 then write('принадлежит') else write('не принадлежит') end. end.</pre>	<pre>INPUT x, y IF x * x + y * y <= 1 THEN IF y >= x THEN IF x <= 0 THEN PRINT "принадлежит" ELSE PRINT "не принадлежит" ENDIF ENDIF ENDIF END</pre>
C++	Алгоритмический язык
<pre>#include <iostream> using namespace std; int main() { float x, y; cin >> x >> y; if (x * x + y * y <= 1) if (y >= x) if (x <= 0) cout << "принадлежит"; else cout << "не принадлежит"; }</pre>	<pre>алг нач вещ x, y ввод x, y если x*x + y*y <= 1 то если y >= x то если x <= 0 то вывод 'принадлежит' иначе вывод 'не принадлежит' все все кон</pre>

Python
<pre> x = float(input()) y = float(input()) if x * x + y * y <= 1: if y >= x: if x <= 0: print("принадлежит") else: print("не принадлежит") </pre>

Последовательно выполните следующее.

1. Перерисуйте и заполните таблицу, которая показывает, как работает программа при аргументах, принадлежащих различным областям (А, В, С, D, E, F, G и H). Точки, лежащие на границах областей, отдельно не рассматривать.



Область	Условие 1 ($x^2 + y^2 \leq 1$)	Условие 2 ($y \geq x$)	Условие 3 ($x \leq 0$)	Программа выведет	Область обрабатывается верно
A					
B					
C					
D					
E					
F					
G					
H					

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, достаточно указать любой способ доработки исходной программы.)

26.1. 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 **удвоить** количество камней в куче. Например, имея кучу из 8 камней, за один ход можно получить кучу из 9 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 42**. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 42 или больше камней.

В начальный момент в куче было S камней: $1 \leq S \leq 41$.

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. Укажите выигрышный ход Пети.

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, причем 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 ребрах дерева указывайте, кто делает ход, в узлах — количество камней в куче.

Решение.

Заметим, что вопросы в задаче построены «по порядку». То есть, вопросы расставлены именно таким образом, чтобы ответ на вопрос 1а можно было использовать при нахождении ответа на вопрос 1б. И так далее. Соответственно настоятельно рекомендуем отвечать на эти вопросы именно в том порядке, в котором они приведены в условии.

Найдём ответ на вопрос 1а. Для победы первым же ходом достаточно получить в куче 42 или более камня. Очевидно, что среди двух возможных ходов $(+1, \times 2)$ ход « $\times 2$ » (удвоение числа камней в куче) изменяет кучу существенно больше, чем ход « $+1$ ». Найдём, какое самое малое количество камней в куче может быть, чтобы одним ходом $(\times 2)$ получить в ней хотя бы 42 камня. Очевидно, это количество — 21. Удвоение числа 21 даёт число 42. Если удваивать большее исходное число, получается ещё больший результат. Значит, наименьшее количество камней в куче для победы первым ходом — 21, наибольшее количество — 41 (наибольшее возможное начальное количество камней в куче).

То есть, ответ на вопрос 1а: $21 \leq S \leq 41$. *Первым ходом Петя должен удвоить число камней в куче.*

Заметим, что в условии не сказано, что нужно указать все возможные выигрышные ходы. Совершенно не обязательно (и даже вредно) указывать отдельно, что в позиции $S = 41$ существует два способа выигрышного хода $(+1$ и $\times 2)$.

Для лучшего понимания следующих действий хорошо бы понять идею *выигрышной* и *проигрышной* позиции игры. Позиция игры называется *выигрышной*, если существует хотя бы один такой ход, что игра сразу же заканчивается победой текущего игрока, или же игра этим ходом переходит в проигрышную позицию. Позиция называется *проигрышной*, если любой ход текущего игрока переводит игру в выигрышную позицию или немедленно заканчивает игру проигрышем текущего игрока.

Заметим, что позиции, найденные в пункте 1а, являются выигрышными. В любой из них существует ход, приводящий текущего игрока непосредственно к выигрышу $(\times 2)$.

Теперь будем искать ответ на вопрос 1б.

Хорошо бы понимать, что требуемая в пункте 1б позиция — проигрышная. Причём такая проигрышная, что после любого хода из неё Петя сразу выиграет одним ходом.

Но все позиции, из которых можно выиграть одним ходом, уже найдены нами в пункте 1а. Значит, нам нужно найти такую позицию, что любой ход из неё приводит число камней в куче в состояние «от 21 до 41». Так как в игре есть ход «+1», единственная такая позиция — $S = 20$ (на 1 меньше, чем граница отрезка из пункта 1а). Действительно, из $S = 20$ можно получить только две позиции ($20 + 1 = 21$ или $20 \times 2 = 40$). Обе они попадают в диапазон из пункта 1а. Значит, именно это нужно указать в ответе:

16. $S = 20$. При любом ходе Пети ($20 + 1 = 21$, $20 \times 2 = 40$) Ваня должен удвоить число камней в куче и выиграть ($21 \times 2 = 42$, $40 \times 2 = 80$).

Найдём ответ на вопрос 2. В данном случае нам требуется найти выигрышную позицию. Причем такую позицию, что она не попадает в список из пункта 1а, и, в то же время, выигрыш случается вторым ходом. Это может быть только в том случае, если после первого хода игра окажется в позиции из пункта 16 ($S = 20$). Получить число 20 можно из двух позиций — из позиции 19 (прибавлением 1) и из позиции 10 (удвоением). Запишем ответ на этот вопрос:

2. $S = 10$ и $S = 19$. В обоих случаях первым ходом Петя должен получить в куче 20 камней ($10 \times 2 = 20$, $19 + 1 = 20$). Независимо от ответного хода Вани ($20 + 1 = 21$, $20 \times 2 = 40$) Петя должен удвоить число камней в куче и выиграть ($21 \times 2 = 42$, $40 \times 2 = 80$).

Заметим, что, хотя в этом пункте можно сослаться на пункт 16 и описать только действия первого хода Пети, рекомендуется так не делать. Во-первых, потому, что если пункт 16 будет как-то не совсем верно описан, то тогда и пункт 2 будет не засчитан. Во-вторых, чтобы не «поощрять привычку». Потому что именно это свойство приводит к тому, что при ответе на пункт 3 экзаменуемые ссылаются на пункт 2. И именно поэтому им не засчитывают пункт 3. В общем, для подстраховки рекомендуем полностью описать ответ на этот вопрос, как приведено выше.

Перейдём к пункту 3. Здесь требуется найти проигрышную позицию. Причём такую позицию, что победа из неё случится на первом или втором ходе другого игрока. И при этом это НЕ позиция из пункта 16 (потому что из неё другой игрок гарантированно выигрывает своим первым ходом).

Нахождение этой позиции имеет смысл делать «начиная» от позиций из пункта 2. Потому что только из позиций пункта 2 игрок выигрывает за 2 хода.

Сначала рекомендуется исследовать позицию, на 1 меньшую большего числа из пункта 2. В данном случае это позиция $S = 18$. Проверим: Из позиции 18 можно одним ходом попасть только в позиции 19 ($18 + 1$) и 36 (18×2). Обе они уже обозначены нами как выигрышные. Позиция 19 — выигрыш в 2 хода (см. пункт 2). Позиция 36 — выигрыш в 1 ход (см. пункт 1а). Значит, наше предположение было верно. Можно записать ответ на вопрос 3.

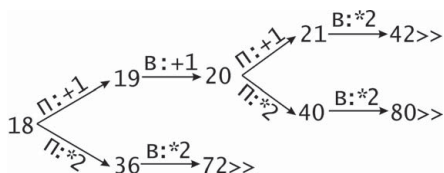
Заметим, что в условии вопроса 3 нас просят, кроме указания позиции, описать выигрышную стратегию Вани и построить дерево всех возможных партий, возможных при этой выигрышной стратегии Вани. Но в критериях оценивания этого задания (с ними можно ознакомиться, например, в конце Демоверсии ЕГЭ по информатике) написано, что этот пункт (3) считается выполненным только если правильно построено именно требуемое дерево.

То есть, если экзаменуемый правильно описал выигрышную стратегию, но при этом не построил требуемое дерево всех возможных партий для этой стратегии, то пункт 3 ему не засчитывается, а если он, наоборот, построил дерево всех возможных партий для выигрышной стратегии, но при этом не написал ни слова о самой стратегии, то пункт 3 ему полностью засчитывается.

Значит, в ответе будем приводить только дерево всех возможных партий для выигрышной стратегии. А саму стратегию описывать не будем для экономии времени.

Приведём ответ на пункт 3:

3. $S = 18$.



Здесь знаком >> обозначены позиции, где партия заканчивается победой Вани.

❗ Важно, чтобы в дереве, которое вы рисуете в качестве ответа, ни в коем случае не были нарисованы ветви, при которых выигрывает Петя! Золотое правило: для выигрывающего игрока рассматриваем только один выигрывающий ход, для проигрывающего игрока — все возможные ходы!

26.1.1. 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 **удвоить** количество камней в куче. Например, имея кучу из 8 камней, за один ход можно получить кучу из 9 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 26**. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **26 или больше** камней.

В начальный момент в куче было S камней, $1 \leq S \leq 25$.

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 обоснуйте 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.

26.1.2. 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 неограни-

ченное 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, $1 \leq S \leq 27$.

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 обоснуйте 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.

Замечание к решению заданий 26.1.3, 26.1.4 и 26.1.5. В этих заданиях имеется три возможных хода игроков. Задания решаются точно так же, как задание 26.1, но при решении не нужно забывать про третий вариант хода. Соответственно, при составлении дерева возможных партий не забудьте у проигрывающего игрока (Пети) рисовать в каждой его позиции все три его возможных хода.

26.1.3. 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 18 камней, за один ход можно получить кучу из 19, 20 или 36 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, $1 \leq S \leq 30$.

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 обоснуйте 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 выигрышающий 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 таких значения S , при которых у Пети есть выигрышная стратегия, причём выполняется два условия: (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 значение S , при котором выполняется два условия:

- 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.

26.1.4. 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, $1 \leq S \leq 30$.

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 обоснуйте 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 выигрышающий 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.

26.1.5. 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 **три** камня или **увеличить** количество камней в куче в **два** раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16, 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 24**. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет **24 или больше** камней.

В начальный момент в куче было S камней, $1 \leq S \leq 23$.

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 обоснуйте 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 выигрышающий 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.

Рекомендации к решению заданий 26.2. В этих заданиях имеется две кучи камней и два возможных хода для каждой кучи. Соответственно, у каждого игрока в каждый момент игры имеется четыре возможных хода (+1 или $\times 2$ к первой куче, и +1 или $\times 2$ ко второй куче). Соответственно, при решении и в ответе-дереве нужно не забывать указывать все 4 возможных варианта хода. Хотя эти задания на первый взгляд выглядят значительно отличающимися от заданий 26.1, решаются они точно так же — последовательно, от вопроса 1 к вопросу 3, пользуясь при нахождении ответа на последующий вопрос результатами предыдущего.

26.2.1. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат 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 (15,40). 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 58**. Победителем считается игрок, сделавший последний ход, то есть первым получивший положение, в котором сумма камней в кучах будет **58 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 (8,25), (7,26), (10,24) 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 (7,25) и (9,24) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

2. Для исходных позиций (6,25), (8,24) и (9,23) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

3. Для исходной позиции (8,23) 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.

26.2.2. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат 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 46**. Победителем считается игрок, сделавший последний ход, то есть первым получивший положение, в котором сумма камней в кучах будет **46 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 (6,20), (5,21), (8,19) 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 (5,20) и (7,19) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

2. Для исходных позиций (5,19), (6,19) и (7,18) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

3. Для исходной позиции (6,18) 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.

26.2.3. 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 четырех 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;

$$1 \leq S \leq 25.$$

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 выигрышающие ходы. Если при некотором значении S Петя может выиграть несколькими способами, достаточно указать один выигрышающий 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 нужно указать одно значение S и объяснить, почему это значение удовлетворяет условию соответствующего задания.

26.2.4. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. Назовем 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.

26.2.5. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. Назовем 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 57**. Победителем считается игрок, сделавший последний ход, то есть первым получивший положение, в котором сумма камней в кучах будет **57 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 (6,24), (7,25), (9,24) 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 (6,25) и (8,24) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

2. Для исходных позиций (6,24), (8,23) и (7,24) укажите, кто из игроков выигрывает. Сколько ходов необходимо для выигрыша? Опишите выигрышную стратегию. Обоснуйте ответ.

3. Для исходной позиции (7,23) 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.

Рекомендации к решению заданий 26.3. В заданиях 26.3.1 и 26.3.2 появляется дополнительное условие. Окончание партии не означает обязательно победы того игрока, который сделал последний ход. Победа зависит от того, сколько камней оказалось в куче. Соответственно, нужно аккуратнее отнестись к поиску решений. Так, например, частая ошибка при ответе на вопрос 1а — «потерять» выигрышную позицию, из которой можно выиграть первым ходом, добавив в кучу 1 камень.

Также важно понимать существенное отличие ответа на пункты 1-2 и на пункт 3. В пунктах 1 и 2 нужно описать выигрышную стратегию. Это можно сделать словами или деревом. И при этом можно (и, как правило, очень хочется) ссылаться на уже описанные выше в решении деревья и стратегии. И ещё — в описании этих стратегий можно не указывать ход проигрывающего игрока, немедленно заканчивающий игру и приводящий его сразу к проигрышу. Потому что описание стратегии (по условию) — значит описать, какой должен быть ход **выигрывающего** игрока в любой позиции, которая ему встретится в игре.

А вот в пункте 3 нужно нарисовать дерево всех возможных партий, возможных при выигрышной стратегии. В этом дереве обязательно должны быть нарисованы, кроме выигрышных ходов выигрывающего игрока также и те ходы проигрывающего игрока, которые приводят к немедленному окончанию партии!

В задании 26.3.3 обратите внимание на необычную формулировку вопроса 1б!

Задания 26.3.4 и 26.3.5 приведены для «расширения кругозора». Их решение является полезным для понимания идеи выигрышных и проигрышных позиций. Нужно строить из стартовой позиции дерево всех возможных позиций (оно поначалу будет сильно разрастаться — по 4 ветви из каждой позиции). В процессе построения дерева нужно анализировать получаемые позиции на предмет «выигрышности/проигрышности», и отбрасывать ветви, про которые становится известно, что дальше игра гарантированно закончится победой Пети или Вани. Рассмотрев крайние ветви, нужно постепенно возвращаться обратно в корню дерева, делая выводы о все более близких к корню позициях, пока не удастся сделать такой вывод про корень. На основании полученных рассуждений нужно будет выбрать из построенного дерева только ту часть, которая приведет к победе выигрывающего игрока.

26.3.1. 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** камень или **увеличить** количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 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 26**. Если при этом в куче оказалось **не более 40** камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 17 камней, и Петя удвоит количество камней в куче, то игра закончится, и победителем будет Ваня. В начальный момент в куче было S камней, $1 \leq S \leq 25$.

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.

1. а) при каких значениях числа S Петя может выиграть в один ход? Укажите все такие значения и соответствующие ходы Пети;

б) у кого из игроков есть выигрышная стратегия при $S = 24, 23, 22$? Опишите выигрышные стратегии для этих случаев.

2. У кого из игроков есть выигрышная стратегия при $S = 12, 11$? Опишите соответствующие выигрышные стратегии.

3. У кого из игроков есть выигрышная стратегия при $S = 10$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах — количество камней в позиции.

26.3.2. Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может **добавить** в кучу **один** камень или **увеличить** количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 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**. Если при этом в куче оказалось **не более 48** камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 17 камней, и Паша удвоит количество камней в куче, то игра закончится, и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 29$.

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

Выполните следующие задания.

1. а) при каких значениях числа S Паша может выиграть в один ход? Укажите все такие значения и соответствующие ходы Паши;

б) у кого из игроков есть выигрышная стратегия при $S = 28, 27, 26$? Опишите выигрышные стратегии для этих случаев.

2. У кого из игроков есть выигрышная стратегия при $S = 14, 13$? Опишите соответствующие выигрышные стратегии.

3. У кого из игроков есть выигрышная стратегия при $S = 12$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах — количество камней в позиции.

26.3.3. 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 четырех 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;

$$1 \leq S \leq 41.$$

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, причем 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.

26.3.4. 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.

26.3.5. 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.

ОТВЕТЫ

№	1.1.1	1.1.2	1.1.3	1.1.4	1.1.5	1.2.1	1.2.2	1.2.3	1.2.4	1.2.5
ОТВЕТ	4	7	5	3	2	6	10	5	9	6
№	1.3.1	1.3.2	1.3.3	1.3.4	1.3.5	1.4.1	1.4.2	1.4.3	1.4.4	1.4.5
ОТВЕТ	2	3	2	3	4	900	56	240	100	448
№	1.5.1	1.5.2	1.5.3	1.5.4	1.5.5	1.6.1	1.6.2	1.6.3	1.6.4	1.6.5
ОТВЕТ	6	10	28	17	89	100	64	16	256	8
№	1.7.1	1.7.2	1.7.3	1.7.4	1.7.5					
ОТВЕТ	153	232	214	190	183					

№	2.1.1	2.1.2	2.1.3	2.1.4	2.1.5	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5
ОТВЕТ	yzx	yxz	xzy	xzy	zyx	yxz	yzx	yzx	xzy	zyx
№	2.3.1	2.3.2	2.3.3	2.3.4	2.3.5	2.4.1	2.4.2	2.4.3	2.4.4	2.4.5
ОТВЕТ	zywx	yzwx	xwzy	xzy	zxy	yxzw	wyxx	wyzx	ywxz	ywxz

№	3.1.1	3.1.2	3.1.3	3.1.4	3.1.5	3.2.1	3.2.2	3.2.3	3.2.4	3.2.5
ОТВЕТ	30	15	20	35	19	4	20	8	19	12
№	3.3.1	3.3.2	3.3.3	3.3.4	3.3.5	3.4.1	3.4.2	3.4.3	3.4.4	3.4.5
ОТВЕТ	16	13	58	48	25	9	11	14	12	9

№	4.1.1	4.1.2	4.1.3	4.1.4	4.1.5	4.2.1	4.2.2	4.2.3	4.2.4	4.2.5
ОТВЕТ	2	5	3	3	3	3474	4218	4218	5363	8208
№	4.3.1	4.3.2	4.3.3	4.3.4	4.3.5					
ОТВЕТ	2	4	2	3	5					

№	5.1.1	5.1.2	5.1.3	5.1.4	5.1.5	5.2.1	5.2.2	5.2.3	5.2.4	5.2.5
ОТВЕТ	00	100	101	1	011	0	A10	C1	A110, B10, C0	D01
№	5.3.1	5.3.2	5.3.3	5.3.4	5.3.5	5.4.1	5.4.2	5.4.3	5.4.4	5.4.5
ОТВЕТ	19	21	18	16	20	5	4	6	6	4

№	6.1.1	6.1.2	6.1.3	6.1.4	6.1.5	6.2.1	6.2.2	6.2.3	6.2.4	6.2.5
ОТВЕТ	B	CE	C	AD	AC	21112	12122	12112	22112	11122
№	6.3.1	6.3.2	6.3.3	6.3.4	6.3.5	6.4.1	6.4.2	6.4.3	6.4.4	6.4.5
ОТВЕТ	DBx	xxD	CxA	BDx	xCD	157	951	960	916	930
№	6.5.1	6.5.2	6.5.3	6.5.4	6.5.5					
ОТВЕТ	54	102	66	78	86					

№	7.1.1	7.1.2	7.1.3	7.1.4	7.1.5	7.2.1	7.2.2	7.2.3	7.2.4	7.2.5
ОТВЕТ	14	4	6	-6	4	=\$C1*2	=D4+1	=\$B2+D\$2	=\$C1-A\$3	=3-A\$5
№	7.3.1	7.3.2	7.3.3	7.3.4	7.3.5	7.4.1	7.4.2	7.4.3	7.4.4	7.4.5
ОТВЕТ	14	6	8	17	13	3	2	4	2	4
№	7.5.1	7.5.2	7.5.3	7.5.4	7.5.5	7.6.1	7.6.2	7.6.3	7.6.4	7.6.5
ОТВЕТ	3	2	4	1	7	6	8	12	18	8

№	8.1.1	8.1.2	8.1.3	8.1.4	8.1.5	8.2.1	8.2.2	8.2.3	8.2.4	8.2.5
Ответ	20	60	127	41	63	1024	256	50	955	23
№	8.3.1	8.3.2	8.3.3	8.3.4	8.3.5	8.4.1	8.4.2	8.4.3	8.4.4	8.4.5
Ответ	33	49	840	720	73	17	39	33	13	32
№	8.5.1	8.5.2	8.5.3	8.5.4	8.5.5					
Ответ	20	35	75	150	240					

№	9.1.1	9.1.2	9.1.3	9.1.4	9.1.5	9.2.1	9.2.2	9.2.3	9.2.4	9.2.5
Ответ	32	1	4	32	256	1	4	2	3	15
№	9.3.1	9.3.2	9.3.3	9.3.4	9.3.5	9.4.1	9.4.2	9.4.3	9.4.4	9.4.5
Ответ	80	60	16	6	10	129	120	1344	10272	2592
№	9.5.1	9.5.2	9.5.3	9.5.4	9.5.5					
Ответ	A39	A37	B10	A24	A14					

№	10.1.1	10.1.2	10.1.3	10.1.4	10.1.5	10.2.1	10.2.2	10.2.3	10.2.4	10.2.5
Ответ	64	4	4	5	243	24	192	62	351	90
№	10.3.1	10.3.2	10.3.3	10.3.4	10.3.5	10.4.1	10.4.2	10.4.3	10.4.4	10.4.5
Ответ	ООЕО	ДДБД	178	299	ТТУСС	120	50	192	120	840
№	10.5.1	10.5.2	10.5.3	10.5.4	10.5.5					
Ответ	324	256	1458	189	108					

№	11.1.1	11.1.2	11.1.3	11.1.4	11.1.5	11.2.1	11.2.2	11.2.3	11.2.4	11.2.5
Ответ	17	14	15	24	25	13	25	12	15	25
№	11.3.1	11.3.2	11.3.3	11.3.4	11.3.5	11.4.1	11.4.2		11.4.3	
Ответ	7	68	64	23	35	12443	135424354		432101210	
№	11.4.4		11.4.5		11.5.1	11.5.2		11.5.3	11.5.4	11.5.5
Ответ	421131211		112111234		252432	51413121		232425	2133	5779

№	12.1.1	12.1.2	12.1.3	12.1.4	12.1.5	12.2.1	12.2.2	12.2.3	12.2.4	12.2.5
Ответ	30	2046	62	126	510	CDBA	CBDA	CADB	CBAD	BCAD
№	12.3.1	12.3.2	12.3.3	12.3.4	12.3.5	12.4.1	12.4.2	12.4.3	12.4.4	12.4.5
Ответ	FCDA	EFCA	CGDA	CDEA	GCDA	240	224	224	240	224
№	12.5.1	12.5.2	12.5.3	12.5.4	12.5.5					
Ответ	19	22	23	20	21					

№	13.1.1	13.1.2	13.1.3	13.1.4	13.1.5	13.2.1	13.2.2	13.2.3	13.2.4	13.2.5
Ответ	420	1400	100	500	125	400	240	32	11	600
№	13.3.1	13.3.2	13.3.3	13.3.4	13.3.5					
Ответ	640	600	160	600	300					

№	14.1.1	14.1.2	14.1.3	14.1.4	14.1.5	14.2.1	14.2.2	14.2.3	14.2.4	14.2.5
Ответ	8877	45544	455	316	275	22777	113333	4415	7388	66116
№	14.3.1	14.3.2	14.3.3	14.3.4	14.3.5	14.4.1	14.4.2	14.4.3	14.4.4	14.4.5
Ответ	23	13	27	23	36	7	3	5	5	7

№	15.1.1	15.1.2	15.1.3	15.1.4	15.1.5	15.2.1	15.2.2	15.2.3	15.2.4	15.2.5
ОТВЕТ	14	17	20	75	46	30	12	36	13	41
№	15.3.1	15.3.2	15.3.3	15.3.4	15.3.5	15.4.1	15.4.2	15.4.3	15.4.4	15.4.5
ОТВЕТ	28	18	28	21	21	455	437	374	308	490

№	16.1.1	16.1.2	16.1.3	16.1.4	16.1.5	16.2.1	16.2.2	16.2.3	16.2.4	16.2.5
ОТВЕТ	7	5	6	28	30	7,14,28	6,8,12,24	6,9,18	5,13,21	7,16,25
№	16.3.1	16.3.2	16.3.3	16.3.4	16.3.5	16.4.1	16.4.2	16.4.3	16.4.4	16.4.5
ОТВЕТ	5	6	333	75	5	46	13	99	52	13
№	16.5.1	16.5.2	16.5.3	16.5.4	16.5.5					
ОТВЕТ	274	316	3C	9	8					

№	17.1.1	17.1.2	17.1.3	17.1.4	17.1.5	17.2.1	17.2.2	17.2.3	17.2.4	17.2.5
ОТВЕТ	3421	3124	4231	8900	2200	1500	2600	16500	13200	11600
№	17.3.1	17.3.2	17.3.3	17.3.4	17.3.5					
ОТВЕТ	11	16	47	29	27					

№	18.1.1	18.1.2	18.1.3	18.1.4	18.1.5	18.2.1	18.2.2	18.2.3	18.2.4	18.2.5
ОТВЕТ	45	2	5	36	24	14	6	5	14	3
№	18.3.1	18.3.2	18.3.3	18.3.4	18.3.5	18.4.1	18.4.2	18.4.3	18.4.4	18.4.5
ОТВЕТ	12	10	34	33	24	19	13	65	99	219

№	19.1.1	19.1.2	19.1.3	19.1.4	19.1.5	19.2.1	19.2.2	19.2.3	19.2.4	19.2.5
ОТВЕТ	22	7	9	8	10	6	5	9	4	4
№	19.3.1	19.3.2	19.3.3	19.3.4	19.3.5	19.4.1	19.4.2	19.4.3	19.4.4	19.4.5
ОТВЕТ	4	5	4	3	7	4	5	9	39	2

№	20.1.1	20.1.2	20.1.3	20.1.4	20.1.5	20.2.1	20.2.2	20.2.3	20.2.4	20.2.5
ОТВЕТ	107	997	555	234	2235	58	337	544	950	206
№	20.3.1	20.3.2	20.3.3	20.3.4	20.3.5	20.4.1	20.4.2	20.4.3	20.4.4	20.4.5
ОТВЕТ	316	173	74	686	631	13	14	210	70	120

№	21.1.1	21.1.2	21.1.3	21.1.4	21.1.5	21.2.1	21.2.2	21.2.3	21.2.4	21.2.5
ОТВЕТ	496	191	134	113	64	63	76	128	511	256
№	21.3.1	21.3.2	21.3.3	21.3.4	21.3.5	21.4.1	21.4.2	21.4.3	21.4.4	21.4.5
ОТВЕТ	20	18	49	12	16	6	10	3	13	4

№	22.1.1	22.1.2	22.1.3	22.1.4	22.1.5	22.2.1	22.2.2	22.2.3	22.2.4	22.2.5
ОТВЕТ	47	28	14	11	7	25	37	44	169	132
№	22.3.1	22.3.2	22.3.3	22.3.4	22.3.5	22.4.1	22.4.2	22.4.3	22.4.4	22.4.5
ОТВЕТ	276	20	70	18	108	56	152	48	45	102

№	23.1.1	23.1.2	23.1.3	23.1.4	23.1.5	23.2.1	23.2.2	23.2.3	23.2.4	23.2.5
ОТВЕТ	243	72	32	32	364	16	16	22	192	108
№	23.3.1	23.3.2	23.3.3	23.3.4	23.3.5	23.4.1	23.4.2	23.4.3	23.4.4	23.4.5
ОТВЕТ	64	96	97	192	127	42	88	55	68	21

№ 24.1.1.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 7</p> <p>2) 128 (или любое натуральное число, в котором либо последняя цифра — наибольшая чётная, либо наибольшая чётная цифра в числе не 0 и больше последней цифры числа)</p> <p>3) (Исправление программы на языке Паскаль) В строке «$m := N \bmod 10;$» должно быть «$m := -1;$», в строке «if $m = 0$ then» должно быть «if $m = -1$ then».</p>	
Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить четыре действия (правильно ответить на вопросы 1 и 2 и исправить две ошибки). То есть:</p> <p>1. Верно указать, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указать число, при котором программа работает верно</p> <p>3–4. Указать и верно исправить две ошибки:</p> <p>а) Указать и верно исправить ошибку инициализации (не обязательно с упоминанием этого термина).</p> <p>б) Указать и верно исправить ошибку проверки на отсутствие чётных цифр в числе.</p> <p>Каждый из п. а) и б) считается выполненным, если:</p> <p>i) правильно указана строка с ошибкой;</p> <p>ii) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</p>	
<p>Правильно выполнены все пункты задания. Обе ошибки исправлены верно. Программа после исправлений для всех натуральных чисел N, не превосходящих 10^9, верно определяет наибольшую чётную цифру.</p> <p>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.</p>	3
<p>2 балла за эту задачу выставляется в случае, если нельзя выставить 3 балла, и при этом выполняется один из двух случаев:</p> <p>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.</p> <p>2) Правильно выполнены все четыре действия. Не более одной верной строки указано как ошибочные.</p>	2
<p>Правильно выполнено только два действия из четырёх. Не важно, сколько верных строк указаны как неверные.</p>	1
<p>Все пункты задания выполнены неверно или отсутствуют.</p>	0
<p>Максимальный балл</p>	3

№ 24.1.2.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) NO</p> <p>2) 123 (или любое натуральное число, в котором есть цифра, большая 0 и меньшая 6, либо в котором нет цифр, меньших 6)</p> <p>3) (Исправление программы на языке Паскаль)</p> <p>В строке «m := 0;» должно быть «m := -1;»,</p> <p>в строке «if m > 0 then» должно быть «if m > -1 then».</p>	
Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить четыре действия (правильно ответить на вопросы 1 и 2 и исправить две ошибки). То есть:</p> <p>1. Верно указать, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указать число, при котором программа работает верно</p> <p>3–4. Указать и верно исправить две ошибки:</p> <p>а) Указать и верно исправить ошибку инициализации (не обязательно с упоминанием этого термина).</p> <p>б) Указать и верно исправить ошибку проверки на отсутствие в числе цифр, меньших 6.</p> <p>Каждый из п. а) и б) считается выполненным, если:</p> <p>i) правильно указана строка с ошибкой;</p> <p>ii) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</p>	
<p>Правильно выполнены все пункты задания.</p> <p>Обе ошибки исправлены верно.</p> <p>Программа после исправлений для всех натуральных чисел N, не превосходящих 10^9, верно определяет наибольшую цифру, меньшую 6. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.</p>	3
<p>2 балла за эту задачу выставляется в случае, если нельзя выставить 3 балла, и при этом выполняется один из двух случаев:</p> <p>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.</p> <p>2) Правильно выполнены все четыре действия. Не более одной верной строки указано как ошибочные.</p>	2
<p>Правильно выполнено только два действия из четырёх. Не важно, сколько верных строк указаны как неверные.</p>	1
<p>Все пункты задания выполнены неверно или отсутствуют.</p>	0
<p>Максимальный балл</p>	3

№ 24.1.3.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 2</p> <p>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)</p> <p>3) (Исправление программы на языке Паскаль) В строке «while N > 1 do» должно быть «while N > 0 do», в строке «writeln(d)» должно быть «writeln(sum)».</p>	
Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить четыре действия (правильно ответить на вопросы 1 и 2 и исправить две ошибки). То есть:</p> <p>1. Верно указать, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указать число, при котором программа работает верно</p> <p>3–4. Указать и верно исправить две ошибки:</p> <p>а) Указать и верно исправить ошибку условия цикла.</p> <p>б) Указать и верно исправить ошибку вывода на экран не той переменной. Каждый из п. а) и б) считается выполненным, если:</p> <p>i) правильно указана строка с ошибкой;</p> <p>ii) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.</p>	
<p>Правильно выполнены все пункты задания.</p> <p>Обе ошибки исправлены верно.</p> <p>Программа после исправлений для всех натуральных чисел N, не превосходящих 10^9, верно определяет сумму нечётных цифр.</p> <p>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.</p>	3
<p>2 балла за эту задачу выставляется в случае, если нельзя выставить 3 балла, и при этом выполняется один из двух случаев:</p> <p>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.</p> <p>2) Правильно выполнены все четыре действия. Не более одной верной строки указано как ошибочные.</p>	2
<p>Правильно выполнено только два действия из четырёх. Не важно, сколько верных строк указаны как неверные.</p>	1
<p>Все пункты задания выполнены неверно или отсутствуют.</p>	0
<p>Максимальный балл</p>	3

№ 24.1.4.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 2</p> <p>2) 5 (или любое однозначное число или число, в котором все цифры одинаковые)</p> <p>3) (Исправление программы на языке Паскаль)</p> <p>В строке «k := 10;» должно быть «k := 0;», в строке «if N mod 10 < k then» должно быть «if N mod 10 > k then».</p>	
Указания по оцениванию	
<p>Обратите внимание!</p> <p>В задаче требовалось выполнить три действия.</p> <p>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.</p> <p>1. Верно указано, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указано число, при котором программа работает верно</p> <p>3. Указаны и верно исправлены две ошибки:</p> <p>а) Указана и верно исправлена ошибка инициализации (не обязательно с упоминанием этого термина).</p> <p>б) Указано на неверное условие выбора максимума, и оно исправлено на верное.</p> <p>Каждый из п. а) и б) считается выполненным, если:</p> <p>i) правильно указана строка с ошибкой;</p> <p>ii) указан такой новый вариант строки, что при исправлении второй ошибки получается правильная программа.</p>	
<p>Правильно выполнены все пункты задания.</p> <p>Обе ошибки исправлены верно. Программа после исправлений для всех натуральных чисел N, не превосходящих 10^9, верно определяет наибольшую цифру.</p> <p>В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.</p>	3
<p>Правильно выполнены два действия из трёх.</p> <p>Верное указание на ошибку при неверном исправлении при этом не засчитывается.</p> <p>Допустимо, чтобы в пункте 3 была исправлена только одна ошибка.</p>	2
<p>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 ее явного указания и исправления или в новом тексте программы</p>	1
<p>Все пункты задания выполнены неверно или отсутствуют.</p>	0
<p>Максимальный балл</p>	3

№ 24.1.5.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 7</p> <p>2) 10 (или любое натуральное число, оканчивающееся на 0)</p> <p>3) (Исправление программы на языке Паскаль)</p> <p>В строке «s := 0;» должно быть «s := N mod 10;».</p>	
Указания по оцениванию	
<p>Обратите внимание!</p> <p>В задаче требовалось выполнить три действия.</p> <p>Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.</p> <p>1. Верно указано, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указано число, при котором программа работает верно</p> <p>3. Указана и верно исправлена ошибка инициализации (не обязательно с упоминанием этого термина).</p> <p>Пункт 3 считается выполненным, если:</p> <p>а) правильно указана строка с ошибкой;</p> <p>б) указан такой новый вариант строки, что при исправлении ошибки получается правильная программа.</p>	
<p>Правильно выполнены все пункты задания.</p> <p>Ошибка исправлена верно. Программа после исправлений для всех натуральных чисел N, не превосходящих 10^9, верно определяет сумму цифр числа. В работе (во фрагментах программ) допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения.</p>	3
<p>Правильно выполнены два действия из трёх.</p> <p>Верное указание на ошибку при неверном исправлении при этом не засчитывается.</p>	2
<p>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</p>	1
<p>Все пункты задания выполнены неверно или отсутствуют.</p>	0
Максимальный балл	3

№ 24.2.1.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 0</p> <p>2) 1 2 3 4 (или любая последовательность 4-х целых чисел, в которой есть хотя бы одно положительное число)</p>	

3) (Исправление программы на языке Паскаль) В строке «if min > 0 then» должно быть «if min < 2000 then» (или min <> 2000 или min <= 1000). В строке «writeln(0)» должно быть «writeln('NO')».	
Указания по оцениванию	
Обратите внимание! В задаче требовалось выполнить четыре действия: 1. Указать, что выведет программа для указанных входных данных, 2. Привести пример входных данных, при которых программа работает верно, 3. Исправить одну ошибку в программе, 4. Исправить вторую ошибку в программе	
Правильно выполнены все пункты задания. Ошибки исправлены верно. 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

№ 24.2.2.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
Элементы ответа: 1) 6 2) 1 2 4 6 (или любая последовательность 4-х целых чисел, в которой нечётные числа — единицы, а остальные числа — чётные) 3) (Исправление программы на языке Паскаль) В строке «if a mod 2 > 0 then» должно быть «if a mod 2 <> 0 then». В строке «count := count + a» должно быть «count := count + 1».	
Указания по оцениванию	
Обратите внимание! В задаче требовалось выполнить четыре действия: 1. Указать, что выведет программа для указанных входных данных, 2. Привести пример входных данных, при которых программа работает верно, 3. Исправить одну ошибку в программе, 4. Исправить вторую ошибку в программе	

Правильно выполнены все пункты задания. Ошибки исправлены верно. 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

№ 24.2.3.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 0</p> <p>2) 0 0 0 0 (или любая последовательность 4-х целых чисел, в которой есть хотя бы один ноль, а другие числа отрицательные)</p> <p>3) (Исправление программы на языке Паскаль) В строке <code>max := 0;</code> должно быть <code>max := -1001</code> (или любое число ≤ 1000). В строке <code>a := max</code> должно быть <code>max := a</code>.</p>	
Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <ol style="list-style-type: none"> 1. Указать, что выведет программа для указанных входных данных, 2. Привести пример входных данных, при которых программа работает верно, 3. Исправить одну ошибку в программе, 4. Исправить вторую ошибку в программе 	
Правильно выполнены все пункты задания. Ошибки исправлены верно. 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

№ 24.2.4.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) «NO»</p> <p>2) 1 2 3 4 (или любая последовательность 4-х целых чисел, у которой сумма чётных чисел больше нуля)</p> <p>3) (Исправление программы на языке Паскаль)</p> <p>В строке «count := 1;» должно быть «count := 0;».</p> <p>В строке «if sum>0 then» должно быть «if count>0 then».</p>	
Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить четыре действия:</p> <p>1. Указать, что выведет программа для указанных входных данных,</p> <p>2. Привести пример входных данных, при которых программа работает верно,</p> <p>3. Исправить одну ошибку в программе,</p> <p>4. Исправить вторую ошибку в программе</p>	
Правильно выполнены все пункты задания. Ошибки исправлены верно. 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

№ 24.2.5.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) –5</p> <p>2) 1 2 3 0 (или любая последовательность 4-х целых чисел, в которой либо нет отрицательных чисел, а последнее число — 0, либо только одно отрицательное число — последнее)</p> <p>3) (Исправление программы на языке Паскаль)</p> <p>В строке «sum := a» должно быть «sum := sum + a».</p> <p>В строке «writeln(a)» должно быть «writeln(sum)».</p>	

Указания по оцениванию	
Обратите внимание! В задаче требовалось выполнить четыре действия: 1. Указать, что выведет программа для указанных входных данных, 2. Привести пример входных данных, при которых программа работает верно, 3. Исправить одну ошибку в программе, 4. Исправить вторую ошибку в программе	
Правильно выполнены все пункты задания. Ошибки исправлены верно. 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

№ 24.2.6.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 3</p> <p><i>Примечание.</i> Программа считает среднее арифметическое чётных элементов. Поэтому результат будет $(2 + 4)/2 = 3$.</p> <p>2) 2 3 4 0</p> <p>Правильным ответом является любая последовательность целых чисел, оканчивающаяся нулём, в которой среднее арифметическое чётных чисел равно среднему арифметическому нечётных чисел. При этом эти средние арифметические не должны равняться нулю.</p> <p>3) В программе две ошибки:</p> <p>а) вместо проверки на нечётность числа проверяются на чётность,</p> <p>б) неверно проверяется отсутствие нечётных чисел. Проверка суммы на ноль не верна, например, для последовательности $-3\ 3\ 0$.</p> <p>Исправление программы на языке Паскаль. Первая ошибка: строку «if a mod 2 = 0 then» заменить на «if a mod 2 <> 0 then». Вторая ошибка: в строке «if s = 0 then» должно быть «if c = 0 then».</p>	

Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить три действия. Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.</p> <p>1. Верно указано, что именно выведет программа при указанных в условии входных данных.</p> <p>2. Указаны входные данные, при которых программа работает верно.</p> <p>3. Указана и верно исправлена ошибка, не проверяющая первое введённое число (но проверяющая последнее введенное число).</p> <p>Пункт 3 считается выполненным, если правильно указано, как именно нужно исправить ошибку.</p>	
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. 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

№ 24.2.7.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)	Баллы
<p>Элементы ответа:</p> <p>1) 30</p> <p><i>Примечание.</i> Программа выводит на экран сумму чисел, кратных пяти.</p> <p>2) 1 2 3 4 (или любая последовательность из 4-х натуральных чисел, не содержащая чисел, кратных пяти)</p> <p>3) (Исправление программы на языке Паскаль)</p> <p>Первая ошибка: строку «if a mod 5 = 0 then» заменить на строку «if a mod 10 = 5 then».</p> <p>Вторая ошибка: строку «s := s + a» заменить на строку «s := s + 1».</p>	
Указания по оцениванию	
<p>Обратите внимание! В задаче требовалось выполнить три действия. Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.</p>	

1. Верно указано, что именно выведет программа при указанных в условии входных данных. 2. Указаны входные данные, при которых программа работает верно. 3. Указана и верно исправлена ошибка, не проверяющая первое введенное число (но проверяющая последнее введенное число). 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 из 4-х натуральных чисел верно определяет количество элементов последовательности, оканчивающихся на 5. 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

№ 24.3.1.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)					Баллы
Элементы ответа:					
1.					
Область	Условие1 ($x \leq 11$)	Условие2 ($x \geq -1$)	Условие3 ($x < 7$)	Программа выведет	Область обрабатывается верно
A	да	нет	—	—	нет
B	да	да	да	принадлежит	да
C	да	да	да	принадлежит	нет
D	да	да	нет	не принадлежит	нет
E	нет	—	—	—	нет
2. Возможная доработка (Паскаль): <pre> if (x>=-1) and (x<=3) or (x>=7) and (x<=11) then write('принадлежит') else write('не принадлежит')</pre>					
Возможны и другие способы доработки.					

Например:

```
if x>=-1 then
  if x<=3 then
    write('принадлежит')
  else
    if x>=7 then
      if x<=11 then
        write('принадлежит')
      else
        write('не принадлежит')
    else
      write('не принадлежит')
else
  write('не принадлежит')
```

Указания по оцениванию

Обратите внимание! В задаче требуется выполнить три действия.

1. Заполнить таблицу.

2. Исправить ошибку в условном операторе.

3. Исправить ошибку, связанную с неправильным набором условий.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.

1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.

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.

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.

3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие $x \leq 3$). 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.

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.

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 « $x \leq 3$ » используется « $x < 3$ »	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

№ 24.3.2.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)					Баллы
Элементы ответа:					
1.					
Область	Условие1 ($x > 1$)	Условие2 ($x = 7$)	Условие3 ($x \leq 13$)	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): <pre> if (x>=-5) and (x<=1) or (x>=7) and (x<=13) then write('принадлежит') else write('не принадлежит') </pre> 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

Обратите внимание! В задаче требуется выполнить три действия.

1. Заполнить таблицу.

2. Исправить ошибку в условном операторе.

3. Исправить ошибку, связанную с неправильным набором условий.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.

1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.

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.

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.

3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие $x \geq -5$). 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.

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.

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 « $x \geq -5$ » 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

№ 24.3.3.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)					Баллы
Элементы ответа:					
1.					
Область	Условие1 ($x \geq 3$)	Условие2 ($x \leq 9$)	Условие3 ($x = 7$)	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):					
<pre> if (x>=3) and (x<=5) or (x>=7) and (x<=9) then write('принадлежит') else write('не принадлежит') </pre>					
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

Обратите внимание! В задаче требуется выполнить три действия.

1. Заполнить таблицу.

2. Исправить ошибку в условном операторе.

3. Исправить ошибку, связанную с неправильным набором условий.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия. Рассмотрим отдельно каждое действие.

1. Действие по заполнению таблицы считается выполненным, если в таблице нет ошибок или ошибки присутствуют только в одной строке.

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.

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.

3. Приведённых трёх ограничений недостаточно для описания двух областей (потеряно условие $x \leq 5$). 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.

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.

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 « $x \leq 5$ » 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

№ 24.3.4.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)					Баллы
Элементы ответа: 1.					
Область	Условие1 ($y \geq \text{abs}(x) - 1$)	Условие2 ($y \leq 0$)	Условие3 ($y \leq -x * x$)	Программа выведет	Область обрабатывается верно
A	нет	—	—	—	нет
B	да	нет	—	—	нет
C	нет	—	—	—	нет
D	нет	—	—	—	нет
E	нет	—	—	—	нет
F	да	да	нет	не принадлежит	нет
G	да	да	да	принадлежит	да
H	нет	—	—	—	нет
I	да	да	нет	не принадлежит	да
2. Возможная доработка (Паскаль): <pre> if (x<=0) and (y>=abs(x)-1) and (y<=0) or (x>=0) and (y>=abs(x)-1) and (y<=-x*x) then write('принадлежит') else write('не принадлежит')</pre>					

<p>Возможны и другие способы доработки.</p> <p>Пример:</p> <pre>if (y>=abs(x)-1) and ((y<=-x*x) or (x<=0) and (y<=0)) then write('принадлежит') else write('не принадлежит')</pre>	
Указания по оцениванию	
<p>Обратите внимание! 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. Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.</p> <p>1. Верное заполнение предложенной таблицы.</p> <p>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.</p> <p>В сложных случаях это действие считается выполненным, если программа выдаёт 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.</p> <p>3. Исправление ошибки, из-за которой приведённым трем ограничениям не удовлетворяют точки плоскости, у которых $y > -x^2$, $y \geq x - 1$, $y \leq 0$ и $x < 0$ (область F). Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области её части.</p> <p>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.</p>	
Правильно выполнены оба пункта задания. Исправлены две ошибки. Программа для всех пар чисел x , y верно определяет принадлежность точки закрашенной области. 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).	2

При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $y \geq -x$ » используется « $y > -x$ ».	
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).	
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. 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

№ 24.3.5.

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)					Баллы
Элементы ответа: 1.					
Область	Условие1 ($x * x + y * y \leq 1$)	Условие2 ($y \geq x$)	Условие3 ($x \leq 0$)	Программа выведет	Область обрабатывается верно
A	нет	—	—	—	нет
B	нет	—	—	—	нет
C	нет	—	—	—	нет
D	нет	—	—	—	нет
E	да	да	да	принадлежит	да
F	да	да	нет	не принадлежит	нет
G	да	нет	—	—	нет
H	да	нет	—	—	нет
2. Возможная доработка (Паскаль):					
<pre> if (x<=0) and (x*x+y*y<=1) or (x>=0) and (x*x+y*y<=1) and (y>=x) then write('принадлежит') else write('не принадлежит') </pre>					
Возможны и другие способы доработки.					

Пример:

```
if (x*x+y*y<=1) and ((x<=0) or (y>=x)) then  
    write('принадлежит')  
else  
    write('не принадлежит')
```

Указания по оцениванию

Обратите внимание! 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.

Баллы за данное задание начисляются как сумма баллов за верное выполнение каждого действия.

1. Верное заполнение предложенной таблицы.

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.

В сложных случаях это действие считается выполненным, если программа выдаёт 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.

3. Исправление ошибки, из-за которой приведённым трем ограничениям не удовлетворяют точки плоскости, у которых $x^2+y^2 \leq 1$ и $y < x$ и $x \leq 0$ (область G), а также те, у которых $x^2+y^2 \leq 1$ и $y \geq x$ и $x > 0$ (область F). Исправлением этой ошибки может быть разбиение области на две части и использование дизъюнкции либо отбрасывание от большей области ее части. 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. Исправлены две ошибки. Программа для всех пар чисел (x, y) верно определяет принадлежность точки закрашенной области.
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).

2

При написании операций сравнения допускается одно неправильное использование строгих/нестрогих неравенств (считается несущественной ошибкой, погрешностью записи). Например, вместо « $y \geq x$ » используется « $y > x$ ».	
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).	
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. 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

№25.1.1

На языке Паскаль	На языке Бейсик
<pre>m := 100; for i:=1 to N do if (a[i]>=40) and (a[i]<m) then m := a[i]; writeln(m)</pre>	<pre>M = 100 FOR I = 1 TO N IF A(I)>=40 AND A(I)<M THEN M = A(I) END IF NEXT I PRINT M</pre>
На языке C++	На алгоритмическом языке
<pre>m = 100; for (i = 0; i < N; i++) if (a[i]>=40 && a[i]<m) m = a[i]; cout << m;</pre>	<pre>М := 100 нц для i от 1 до N если a[i] >= 40 и a[i] < М то М := a[i] все кц вывод М</pre>
На языке Python	
<pre>m = 100 for i in range(0, n): if a[i] >= 40 and a[i] < m: m = a[i] print(m)</pre>	

№ 25.1.2.

На языке Паскаль	На языке Бейсик
<pre> 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]) </pre>	<pre> 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 </pre>
На языке C++	На алгоритмическом языке
<pre> 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]; } </pre>	<pre> 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 </pre>

На языке Python
<pre> 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]) </pre>

№ 25.1.3.

На языке Паскаль	На языке Бейсик
<pre> 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); </pre>	<pre> 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 </pre>
На языке C++	На алгоритмическом языке
<pre> i = 0; while(i < N-1 && a[i] != x) i++; if(a[i] != x) cout << "нет таких"; else cout << i + 1; </pre>	<pre> 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 </pre>
На языке Python	
<pre> i = 0 while i < n - 1 and a[i] != x: i = i + 1 if a[i] != x: print("нет таких") else: print(i + 1) </pre>	

№ 25.1.4.

На языке Паскаль	На языке Бейсик
<pre> 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')</pre>	<pre> 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</pre>
На языке C++	На алгоритмическом языке
<pre> 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";</pre>	<pre> m := 1000 <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N <u>если</u> a[i]>99 <u>и</u> a[i]<1000 <u>и</u> mod(a[i],10)<>4 <u>и</u> a[i]<m <u>то</u> m := a[i] <u>все</u> <u>кц</u> <u>если</u> m = 1000 <u>то</u> <u>вывод</u> 'NO' <u>иначе</u> <u>вывод</u> m <u>все</u></pre>
На языке Python	
<pre> 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")</pre>	

№ 25.1.5.

На языке Паскаль	На языке Бейсик
<pre> k := 0; for i := 1 to N do if (a[i]>=100) and (a[i]<=999) then k := k + 1; if k > 0 then writeln(k) else writeln('NO') </pre>	<pre> K = 1000 FOR I = 1 TO N IF (A(I)>99) AND (A(I)<1000) THEN K = K + 1 END IF NEXT I IF K = 0 THEN PRINT "NO" ELSE PRINT K END IF </pre>
На языке C++	На алгоритмическом языке
<pre> k = 0; for(i = 0 ; i < N ; i++) if(a[i]>99 && a[i]<1000) k++; if(k > 0) cout << k; else cout << "NO"; </pre>	<pre> к := 1000 <u>нц</u> <u>для</u> <u>i</u> <u>от</u> 1 <u>до</u> N <u>если</u> a[i] > 99 <u>и</u> a[i] < 1000 <u>то</u> к := к + 1 <u>все</u> <u>кц</u> <u>если</u> к = 0 <u>то</u> <u>вывод</u> 'NO' <u>иначе</u> <u>вывод</u> к <u>все</u> </pre>
На языке Python	
<pre> k = 0 for i in range(0, n): if a[i] >= 100 and a[i] <= 999: k = k + 1 if k > 0: print(k) else: print("NO") </pre>	

№ 25.2.1

На языке Паскаль	На языке Бейсик
<pre> 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]; writeln(m) </pre>	<pre> 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 PRINT M </pre>

На языке C++	На алгоритмическом языке
<pre> 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]; cout<<m; </pre>	<pre> 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 вывод m </pre>
На языке Python	
<pre> 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] print(m) </pre>	

№ 25.2.2.

На языке Паскаль	На языке Бейсик
<pre> 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) </pre>	<pre> 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 </pre>
На языке C++	На алгоритмическом языке
<pre> 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; </pre>	<pre> 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 </pre>
На языке Python	
<pre> 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) </pre>	

№ 25.2.3.

На языке Паскаль	На языке Бейсик
<pre>m := 20001; for i:=1 to N div 2 do if a[2*i-1]+a[2*i]<m then m := a[2*i-1]+a[2*i]; writeln(m)</pre>	<pre>M = 20001 FOR I = 1 TO N \ 2 IF A(2*I-1)+A(2*I)<M THEN M := A(2*I-1)+A(2*I) END IF NEXT I PRINT M</pre>
На языке C++	На алгоритмическом языке
<pre>m = 20001; for(i = 0 ; i < N / 2 ; i++) if(a[2*i]*a[2*i+1]<m) m = a[2*i]*a[2*i+1]; cout << m;</pre>	<pre>м := 20001 нц для i от 1 до div(N,2) если a[2*i-1]*a[2*i]<m то м := a[2*i-1]*a[2*i] все кц вывод м</pre>
На языке Python	
<pre>m = 20001 for i in range(0, n // 2): if a[2*i] * a[2*i+1] < m: m = a[2*i] * a[2*i+1] print(m)</pre>	

№ 25.2.4.

На языке Паскаль	На языке Бейсик
<pre>k := 0; for i := 1 to N - 1 do if a[i] <= a[i+1] then k := k + 1; if k = 0 then writeln('YES') else writeln('NO')</pre>	<pre>K = 0 FOR I = 1 TO N-1 IF A(I) > A(I+1) THEN K = K + 1 END IF NEXT I IF K = N-1 THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>
На языке C+	На алгоритмическом языке
<pre>k = 0; for(i = 1 ; i < N ; i++) if(a[i-1] <= a[i]) k++; if(k == 0) cout << "YES"; else cout << "NO";</pre>	<pre>к := 0 нц для i от 2 до N если a[i-1] > a[i] то к := к + 1 все кц если k = N-1 то вывод 'YES' иначе вывод 'NO' все</pre>

На языке Python
<pre> k = 0 for i in range(1, n): if a[i-1] <= a[i]: k = k + 1 if k == 0: print("YES") else: print("NO") </pre>

№ 25.2.5.

На языке Паскаль	На языке Бейсик
<pre> max1 := a[1]; max2 := a[2]; if max1 < max2 then begin max1 := a[2]; max2 := a[1] end; for i:=3 to N do if a[i] > max1 then begin max2 := max1; max1 := a[i] end else if a[i] > max2 then max2 := a[i]; writeln(max2) </pre>	<pre> MAX1 = A(1) MAX2 = A(2) IF MAX1 < MAX2 THEN MAX1 = A(2) MAX2 = A(1) END IF FOR I = 3 TO N IF A(I) > MAX1 THEN MAX2 = MAX1 MAX1 = A(I) ELSE IF A(I) > MAX2 THEN MAX2 = A(I) END IF END IF NEXT I PRINT MAX2 </pre>
На языке C++	На алгоритмическом языке
<pre> max1 = a[0]; max2 = a[1]; if(max1 < max2) { max1 = a[1]; max2 = a[0]; } for(i = 2 ; i < N ; i++) if(a[i] > max1) { max2 = max1; max1 = a[i]; } else if(a[i] > max2) max2 = a[i]; cout << max2; </pre>	<pre> MAX1 := a[1] MAX2 := a[2] <u>если</u> MAX1 < MAX2 <u>то</u> MAX1 := a[2] MAX2 := a[1] <u>все</u> <u>нц для</u> i <u>от</u> 3 <u>до</u> N <u>если</u> a[i] > MAX1 <u>то</u> MAX2 := MAX1 MAX1 := a[i] <u>иначе</u> <u>если</u> a[i] > MAX2 <u>то</u> MAX2 := a[i] <u>все</u> <u>все</u> <u>кц</u> <u>вывод</u> MAX2 </pre>

На языке Python
<pre> max1, max2 = a[0], a[1] if max1 < max2: max1, max2 = a[1], a[0] for i in range(2,n): if a[i] > max1: max2, max1 = max1, a[i] else: if a[i] > max2: max2 = a[i] print(max2) </pre>

№ 25.3.1

На языке Паскаль	На языке Бейсик
<pre> i := 1; while (i<N) and (a[i]>=0) do i := i + 1; if a[i] >= 0 then writeln('нет таких') else writeln(i); </pre>	<pre> I = 1 WHILE I < N AND A(I) >= 0 I = I + 1 ENDW IF A(I) >= 0 THEN PRINT "нет таких" ELSE PRINT I END IF </pre>
На языке C++	На алгоритмическом языке
<pre> i = 0; while(i < N-1 && a[i] >= 0) i++; if(a[i] >= 0) cout << "нет таких"; else cout << i+1; </pre>	<pre> i := 1 нц пока i < N и a[i] >= 0 i := i+1 кц если a[i] >= 0 то вывод 'нет таких' иначе вывод i все </pre>
На языке Python	
<pre> i = 0 while i < n-1 and a[i] >= 0: i = i + 1 if a[i] >= 0: print("нет таких") else: print(i+1) </pre>	

№ 25.3.2.

На языке Паскаль	На языке Бейсик
<pre>sum := 0; for i := 1 to n do if (a[i] < 300) and (a[i] mod 10 <> 5) then sum := sum + a[i]; for i := 1 to n do if (a[i] < 300) and (a[i] mod 10 <> 5) then a[i] := sum; for i := 1 to n do writeln(a[i], ' ')</pre>	<pre>SUM = 0 FOR I = 1 TO N IF A(I) < 300 AND A(I) MOD 10<>5 THEN SUM = SUM + A(I) END IF NEXT I FOR I = 1 TO N IF A(I) < 300 AND A(I) MOD 10<>5 THEN A(I) = SUM END IF PRINT A(I) NEXT I</pre>
На языке C++	На алгоритмическом языке
<pre>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; }</pre>	<pre>sum := 0 нц <u>для</u> i <u>от</u> 1 <u>до</u> N <u>если</u> a[i]<300 <u>и</u> mod(a[i],10)<>5 <u>то</u> sum := sum + a[i] <u>все</u> кц нц <u>для</u> i <u>от</u> 1 <u>до</u> N <u>если</u> a[i]<300 <u>и</u> mod(a[i],10)<>5 <u>то</u> a[i] := sum <u>все</u> <u>вывод</u> a[i], <u>нс</u> кц</pre>
На языке Python	
<pre>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])</pre>	

№ 25.3.3.

На языке Паскаль	На языке Бейсик
<pre> k := 0; sum := 0; for i := 1 to n do if (a[i] >= 10) and (a[i] <= 99) then begin k := k + 1; sum := sum + a[i] end; avg := sum / k; for i := 1 to n do if a[i] > avg then write(i, ' '); </pre>	<pre> K = 0 SUM = 0 FOR I = 1 TO N IF A(I) >= 10 AND A(I) <= 99 THEN K = K + 1 SUM = SUM + A(I) END IF NEXT I AVG = SUM / K FOR I = 1 TO N IF A(I) > AVG THEN PRINT I END IF NEXT I </pre>
На языке C++	На алгоритмическом языке
<pre> k = 0; sum = 0; for(i = 0 ; i < N ; i++) if(a[i]>=10 && a[i]<=99) { k++; sum += a[i]; } avg = (double)sum / k; for(i = 0 ; i < N ; i++) if(a[i] > avg) cout << i + 1 << endl; </pre>	<pre> k := 0 sum := 0 нц для i от 1 до N если a[i]>=10 и a[i]<=99 то k := k + 1 sum := sum + a[i] все кц avg := sum / k нц для i от 1 до N если a[i] > avg то вывод i, нс все кц </pre>
На языке Python	
<pre> k = 0 sum = 0 for i in range(0, n): if a[i] >= 10 and a[i] <= 99: k = k + 1 sum = sum + a[i] avg = (sum+0.0) / k for i in range(0, n): if a[i] > avg: print(i + 1) </pre>	

№ 25.3.4.

На языке Паскаль	На языке Бейсик
<pre> m := -10001; for i := 1 to n do if (a[i] mod 3 <> 0) and (a[i] > m) then m := a[i]; for i := 1 to n do if a[i] > m then write(a[i], ' '); </pre>	<pre> M = -10001 FOR I = 1 TO N IF A(I) MOD 3 <> 0 AND A(I) > M THEN M = A(I) END IF NEXT I FOR I = 1 TO N IF A(I) > M THEN PRINT A(I) END IF NEXT I </pre>
На языке C++	На алгоритмическом языке
<pre> m = -10001; for(i=0 ; i<N ; i++) if(a[i]%3!=0 && a[i]>m) m = a[i]; for(i=0 ; i<N ; i++) if(a[i] > m) cout << a[i] << endl; </pre>	<pre> м := -10001 <u>нц</u> <u>для</u> <u>i</u> <u>от</u> 1 <u>до</u> N <u>если</u> mod(a[i],3)<>0 <u>и</u> a[i]>m <u>то</u> м := a[i] <u>все</u> <u>кц</u> <u>нц</u> <u>для</u> <u>i</u> <u>от</u> 1 <u>до</u> N <u>если</u> a[i] > м <u>то</u> <u>вывод</u> a[i], <u>нс</u> <u>все</u> <u>кц</u> </pre>
На языке Python	
<pre> m = -10001 for i in range(0, n): if a[i] % 3 != 0 and a[i] > m: m = a[i] for i in range(0, n): if a[i] > m: print(a[i]) </pre>	

№ 25.3.5.

На языке Паскаль	На языке Бейсик
<pre> m := 101; for i := 1 to n do if (abs(a[i]) <= 100) and (a[i] mod 4 = 0) and (a[i] < m) then m := a[i]; for i := 1 to n do if not ((abs(a[i])<=100) and (a[i] mod 4=0)) then a[i] := m; for i := 1 to n do write(a[i], ' '); </pre>	<pre> M = 101 FOR I = 1 TO N IF ABS(A(I)) <= 100 AND A(I) MOD 4 = 0 and A(I) < M THEN M = A(I) END IF NEXT I FOR I = 1 TO N IF NOT(ABS(A(I))<=100 AND A(I) MOD 4=0) THEN A(I) = M END IF PRINT A(I) NEXT I </pre>
На языке C++	На алгоритмическом языке
<pre> 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; } </pre>	<pre> 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 </pre>
На языке Python	
<pre> 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]) </pre>	

№ 25.4.1

На языке Паскаль	На языке Бейсик
<pre> k := 0; for i := 1 to N do if a[i] > 0 then begin k := k + 1; if k = 3 then j := i end; if k < 3 then writeln('нет такого') else writeln(j) </pre>	<pre> K = 0 FOR I = 1 TO N IF A(I)>0 THEN K = K + 1 IF K = 3 THEN J = I END IF END IF NEXT I IF K<3 THEN PRINT "нет такого" ELSE PRINT J END IF </pre>
На языке C++	На алгоритмическом языке
<pre> k = 0; for(i = 0 ; i < N ; i++) if(a[i] > 0) { k++; if(k == 3) j = i; } if(k < 3) cout << "нет такого"; else cout << j + 1; </pre>	<pre> к := 0 нц для i от 1 до N если a[i] > 0 то к := к + 1 если к = 3 то j := i все все кц если к < 3 то вывод 'нет такого' иначе вывод j все </pre>
На языке Python	
<pre> k = 0 for i in range(0,n): if a[i] > 0: k = k + 1 if k == 3: j = i if k < 3: print("нет такого") else: print(j + 1) </pre>	

№ 25.4.2.

На языке Паскаль	На языке Бейсик
<pre>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)</pre>	<pre>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</pre>
На языке C++	На алгоритмическом языке
<pre>p = 1; for(i = 0 ; i < N ; i++) if(a[i] % 2 != 0 && a[i] > 0) p *= a[i]; cout << p;</pre>	<pre>p := 1 <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N <u>если</u> mod(a[i],2)<>0 <u>и</u> a[i]>0 <u>то</u> p := p * a[i] <u>все</u> <u>кц</u> <u>вывод</u> p</pre>
На языке Python	
<pre>p = 1 for i in range(0, n): if a[i] % 2 != 0 and a[i] > 0: p = p * a[i] print(p)</pre>	

№ 25.4.3.

На языке Паскаль	На языке Бейсик
<pre>k := 0; for i:=1 to N do if (a[i] > 0) and (a[i] mod 2 = 0) then k := k + 1; if k = n then writeln('YES') else writeln('NO')</pre>	<pre>K = 0 FOR I = 1 TO N IF A(I) <= 0 OR A(I) MOD 2 <> 0 THEN K = K + 1 END IF NEXT I IF K = 0 THEN PRINT "YES" ELSE PRINT "NO" END IF</pre>

На языке C++	На алгоритмическом языке
<pre> k = 0; for(i = 0 ; i < N ; i++) if(a[i] > 0 && a[i] % 2 == 0) k++; if(k == n) cout << "YES"; else cout << "NO"; </pre>	<pre> k := 0 нц для i от 1 до N если a[i] > 0 и mod(a[i],2)=0 то k := k + 1 все кц если k = n то вывод 'YES' иначе вывод 'NO' все </pre>
На языке Python	
<pre> k = 0 for i in range(0, n): if a[i] > 0 and a[i] % 2 == 0: k = k + 1 if k == n: print("YES") else: print("NO") </pre>	

№ 25.4.4.

На языке Паскаль	На языке Бейсик
<pre> 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('=') </pre>	<pre> 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 </pre>

На языке C++	На алгоритмическом языке
<pre> 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 << "="; </pre>	<pre> 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 </pre>
На языке Python	
<pre> 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("=") </pre>	

№ 25.4.5.

На языке Паскаль	На языке Бейсик
<pre> 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') </pre>	<pre> 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 </pre>

На языке C++	На алгоритмическом языке
<pre> k = 0; for(i = 0 ; i < N ; i++) if(a[i]>9 && a[i]<100) k++; if(k > 0) cout << "YES"; else cout << "NO"; </pre>	<pre> 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 </pre>
На языке Python	
<pre> 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") </pre>	

№26.1.1. Содержание верного ответа (допускаются иные формулировки ответа, не искажающие его смысла)

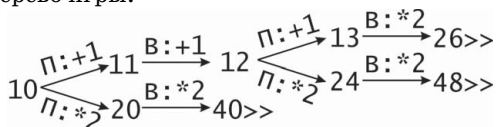
1а. При $S=13\dots25$. Во всех этих случаях Петя должен удвоить количество камней в куче и выиграть. При значениях $S<13$ невозможно одним ходом (+1 или $\cdot 2$) получить 26 или больше камней.

1б. При $S=12$. Куда бы ни пошёл Петя ($12+1=13$ или $12\cdot 2=24$), Ваня удвоит число камней в куче и выиграет ($13\cdot 2=26$, $24\cdot 2=48$).

2. $S=11$ или $S=6$. В обоих случаях Петя может получить в кучке 12 камней ($11+1=12$, $6\cdot 2=12$). При любом ответном ходе Вани ($12+1=13$ или $12\cdot 2=24$) Петя должен удвоить число камней в куче и выиграть ($13\cdot 2=26$, $24\cdot 2=48$).

3. $S=10$. Если Петя удвоит число камней в куче ($10\cdot 2=20$), Ваня тоже должен удвоить число камней в куче и выиграть ($20\cdot 2=40$). Если Петя добавит 1 камень в кучу ($10+1=11$), Ваня также должен добавить 1 камень в кучу ($11+1=12$).

В ответ на любой следующий ход Пети ($12+1=13$ или $12\cdot 2=24$) Ваня должен удвоить число камней в куче и выиграть ($13\cdot 2=26$, $24\cdot 2=48$). 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.

№26.1.2

Содержание верного ответа и указания к оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

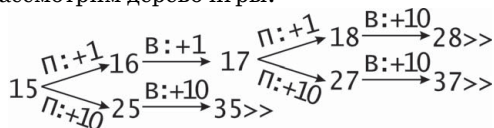
1а. При $S=18...27$. Во всех этих случаях Петя должен добавить в кучу 10 камней и выиграть. При значениях $S<18$ невозможно одним ходом (+1 или +10) получить 28 или больше камней.

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.

№26.1.3

Содержание верного ответа и указания к оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

1а) $S=16...30$. Петя должен увеличить количество камней в куче в 2 раза и выиграть. Для $S<16$ невозможно одним ходом (+1, +2 или *2) получить в куче 31 или больше камней.

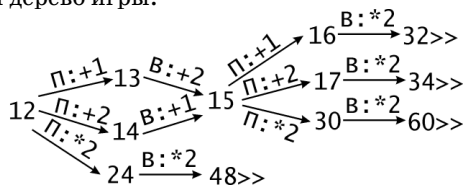
1б) $S=15$. Куда бы ни походил Петя ($15+1=16$, $15+2=17$ или $15*2=30$), Петя должен увеличить количество камней в куче в 2 раза и выиграть ($16*2=32$, $17*2=34$, $30*2=60$).

2. $S=13$ и $S=14$. Своим первым ходом Петя должен получить в куче 15 камней ($13+2=15$ или $14+1=15$). При любом ответном ходе Вани ($15+1=16$, $15+2=17$ или $15*2=30$) Ваня должен увеличить количество камней в куче в 2 раза и выиграть ($16*2=32$, $17*2=34$, $30*2=60$).

3. $S=12$. Если Петя удвоит количество камней в куче ($12*2=24$), Ваня тоже удвоит количество камней в куче и выиграет ($24*2=48$).

Если же Петя добавит в кучу 1 камень или 2 камня ($12+1=13$ или $12+2=14$), Ваня должен довести количество камней в куче до 15 ($13+2=15$ или $14+1=15$). При любом ответном ходе Пети ($15+1=16$, $15+2=17$ или $15*2=30$) Ваня должен увеличить количество камней в куче в 2 раза и выиграть ($16*2=32$, $17*2=34$, $30*2=60$).

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.

№26.1.4

Содержание верного ответа и указания к оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

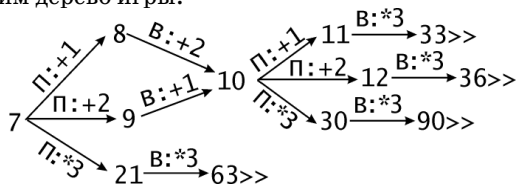
1а. При $S=11\dots30$. Во всех этих случаях Петя должен увеличить количество камней в куче в три раза и выиграть. При значениях $S<11$ невозможно одним ходом (+1, +2 или *3) получить 31 или больше камней.

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:



Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.

№26.1.5

Содержание верного ответа и указания к оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

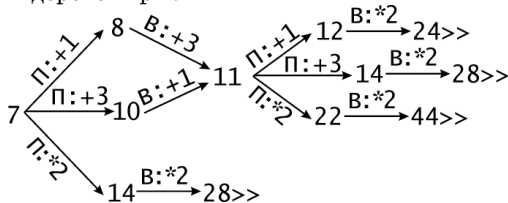
1а. При $S=12\dots23$. Во всех этих случаях Петя должен увеличить количество камней в куче в два раза и выиграть. При значениях $S<12$ невозможно одним ходом (+1, +3 или *2) получить 24 или больше камней.

16. При $S=11$. Куда бы ни пошел Петя ($11+1=12$, $11+3=14$ или $11*2=22$), Ваня удвоит число камней в куче и выиграет ($12*2=24$, $14*2=28$ или $22*2=44$).

2. $S=8$ или $S=10$. В обоих случаях Петя должен получить в кучке 11 камней ($8+3=11$, $10+1=11$). При любом ответном ходе Вани ($11+1=12$, $11+3=14$ или $11*2=22$) Петя должен удвоить число камней в куче и выиграть ($12*2=24$, $14*2=28$ или $22*2=44$).

3. $S=7$ (также верный ответ $S=9$). Если Петя добавит в кучу 1 или 3 камня ($7+1=8$ или $7+3=10$), Ваня должен получить в куче 11 камней ($8+3=11$, $10+1=11$) и в ответ на любой ход Пети ($11+1=12$, $11+3=14$ или $11*2=22$) Ваня должен удвоить число камней в куче и выиграть ($12*2=24$, $14*2=28$ или $22*2=44$). Если Петя удвоит число камней в куче ($7*2=14$), Ваня также должен удвоить число камней в куче и выиграть ($14*2=28$).

Рассмотрим дерево игры:



Дерево всех партий, возможных при Ваниной стратегии. Знаком «>>» обозначены позиции, в которых партия заканчивается.

№26.2.1

Содержание верного ответа и указания к оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

1. Для обеих позиций (7,25) и (9,24) 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 (7,25) Петя может получить позиции (8,25), (14,25), (7,26) и (7,50). Во всех случаях сумма камней в кучах меньше 58. Однако при ходе Вани (удвоение второй кучи) сумма становится не менее 58.

Из позиции (9,24) Петя может получить позиции (10,24), (18,24), (9,25) и (9,48). Во всех случаях сумма камней в кучах меньше 58. Однако при ходе Вани (удвоение второй кучи) сумма становится не менее 58.

2. Для всех трех позиций (6,25), (8,24) и (9,23) выигрышную стратегию имеет Петя. Из позиции (6,25) Петя может добавить 1 камень в первую кучу и получить (7,25). А из позиций (8,24) и (9,23) Петя может получить позицию (9,24), добавив камень во вторую ($8+1,24$) или первую ($9,23+1$) кучу соответственно. Для полученных позиций (7,25) и (9,24) в пункте 1 доказано, что они проигрышные. Как бы ни походил Ваня, Петя удвоит количество камней во второй куче и выиграет.

3. В позиции (8,23) выигрышную стратегию имеет Ваня.

Если Петя первым ходом добавит 1 камень в какую-нибудь кучу, получится позиция (9,23) или (8,24). В пункте 2 доказано, что эти по-