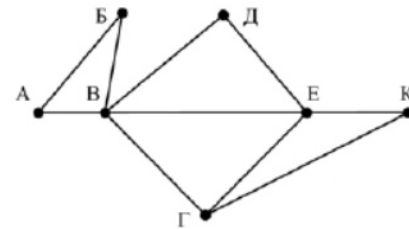


Вариант № 9169595

1. Задание 1 № 9354

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

	П1	П2	П3	П4	П5	П6	П7
П1		45		10			
П2	45			40		55	
П3					15	60	
П4	10	40				20	35
П5			15			55	
П6		55	60	20	55		45
П7				35		45	



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

**Решение.**

Пункт В – единственный пункт с пятью дорогами, значит, ему соответствует П6, а пункт Е – единственный с четырьмя дорогами, значит, ему соответствует П4.

Длина дороги из П6 в П4 равна 20.

Ответ: 20.

Ответ: 20

2. Задание 2 № 16377

Логическая функция  $F$  задаётся выражением  $((x \rightarrow y) \equiv (y \rightarrow z)) \wedge (y \vee w)$ .

Дан частично заполненный фрагмент, содержащий  **неповторяющиеся**  строки таблицы истинности функции  $F$ .

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

Переменная 1	Переменная 2	Переменная 3	Переменная 4	Функция
???	???	???	???	$F$
0		0		1
0	0		0	1
			0	1

В ответе напишите буквы  $x, y, z, w$  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. Пусть задано выражение  $x \rightarrow y$ , зависящее от двух переменных  $x$  и  $y$ , и фрагмент таблицы истинности:

Переменная 1	Переменная 1	Функция
???	???	$F$

0	1	0
---	---	---

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

**Решение.**

Рассмотрим данное выражение. Преобразуем логическое выражение  $((x \rightarrow y) \equiv (y \rightarrow z)) \wedge (y \vee w)$  и получим систему, при которой оно истинно:

$$\begin{cases} y = 1, \\ w = 1, \\ \bar{x} + y \equiv \bar{y} + z \end{cases} \quad (*)$$

Заметим, что четвёртый столбец таблицы истинности это  $y$ , тогда третий столбец таблицы истинности это переменная  $w$ . Из условия  $\bar{x} + y \neq \bar{y} + z$  следует, что переменная  $x$  соответствует первому столбцу таблицы истинности, а переменная  $z$  соответствует второму столбцу таблицы истинности.

Ответ: xzwy.

**Примечание.**

Рассмотрим, как будет выглядеть полная таблица истинности. Одна из переменных  $y$  или  $w$  должна принимать значение 1, поэтому в третьем столбце во второй и третьей строках будут стоять единицы, и в четвёртом столбце в первой строке будет стоять единица. Исходя из условия  $\bar{x} + y \neq \bar{y} + z$  можно заключить, что в первом столбце в последней строке будет стоять ноль, а в первой строке второго столбца будет стоять единица. В последней строке второго столбца должна стоять единица, поскольку строки в таблице истинности должны быть разными.

Перем. 1	Перем. 2	Перем. 3	Перем. 4
???	???	???	???
0	1	0	1
0	0	1	0
0	1	1	0

Вариант xzwy не подходит, поскольку во второй строке функция F окажется ложной.

**Приведем другое решение.**

Составим таблицу истинности для выражения  $((x \rightarrow y) \equiv (y \rightarrow z)) \wedge (y \vee w)$  и выпишем те наборы переменных, при которых данное выражение равно 1. В наборах переменные запишем в порядке  $x, y, z, w$ . Получим следующие наборы:

- (0, 0, 0, 1),
- (0, 0, 1, 1),
- (0, 1, 1, 0),
- (0, 1, 1, 1),
- (1, 1, 1, 0),
- (1, 1, 1, 1).

Сопоставим эти наборы с приведенным в задании фрагментом таблицы истинности.

Вторая строка таблицы истинности содержит три нуля, следовательно, она может соответствовать только набору (0, 0, 0, 1), тогда в третьем столбце второй строки стоит 1, и третий столбец соответствует переменной  $w$ .

Рассмотрим первую строку таблицы. В ней переменная  $w$  принимает нулевое значение, и есть еще одна переменная, принимающая нулевое значение, значит, эта строка может соответствовать только набору (0, 1, 1, 0), тогда первый столбец соответствует переменной  $x$ .

Рассмотрим третью строку таблицы. В ней одна из переменных  $y$  или  $z$  принимает нулевое значение, значит, эта строка может соответствовать только набору (0, 0, 1, 1), и четвертый столбец соответствует переменной  $y$ , тогда второй столбец соответствует переменной  $z$ .

Получаем xzwy.

Ответ: xzwy

3. Задание 3 № 5044

В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите ID родного брата Седых В. А.

Таблица 1			Таблица 2	
ID	Фамилия_И._О.	Пол	ID_Родителя	ID_Ребенка
1588	Саенко М. А.	Ж	1616	1588
1616	Билич А. П.	М	2349	1588
1683	Виктюк И. Б.	М	2008	1683
1748	Кеосаян А. И.	Ж	2106	1683
1960	Виктюк П. И.	М	1683	1960
1974	Тузенбах П. А.	Ж	2882	1960
2008	Виктюк Б. Ф.	М	2860	1974
2106	Чирик Д. К.	Ж	2860	2339
2339	Седых Л. А.	М	2008	2349
2349	Виктюк А. Б.	Ж	2106	2349
2521	Меладзе К. Г.	М	1616	2593
2593	Билич П. А.	М	2349	2593
2730	Виктюк Т. И.	Ж	1683	2730
2860	Панина Р. Г.	Ж	2882	2730
2882	Шевченко Г. Р.	Ж	1616	2911
2911	Седых В. А.	Ж	2349	2911

**Решение.**

Из первой таблицы определяем, что ID Седых В.А. — 2911.

Из второй таблицы определяем, что родители Седых В.А. — 1616, 2349.

Из второй таблицы определяем, что второй ребенок 1616 и 2349 — 2593, 1588.

Из первой таблицы определяем, что 1588 и 2593 — Саенко М.А.(ж) и Билич П.А.(м).

Ответ: 2593.

Ответ: 2593



#### 4. Задание 4 № 3673

Для 5 букв латинского алфавита заданы их двоичные коды (для некоторых букв — из двух бит, для некоторых — из трех). Эти коды представлены в таблице:

a	b	c	d	e
100	110	011	01	10

Какой набор букв закодирован двоичной строкой 1000110110110? Все буквы в последовательности — разные.

**Решение.**

Мы видим, что условия Фано и обратное условие Фано не выполняются, значит, код можно раскодировать неоднозначно.

Будем пробовать разные варианты, отбрасывая те, в которых получаются повторяющиеся буквы:

1) 100 011 01 10 110

Первая буква определяется однозначно, её код 100: a.

Пусть вторая буква — c, тогда следующая буква — d, потом — e и b.

Такой вариант удовлетворит условию, значит, окончательно получили ответ: **acdeb**.

Ответ: acdeb

#### 5. Задание 5 № 3396

Исполнитель Вычислитель работает с целыми положительными однобайтными числами. Он может выполнять две команды:

1. сдвинь биты числа влево на одну позицию
2. прибавь 1

Например, число 7 ( $00000111_2$ ) преобразуется командой 1 в 14 ( $00001110_2$ ). Для заданного числа 14 выполнена последовательность команд 11222. Запишите полученный результат в десятичной системе счисления.

**Решение.**

Если в старшем разряде нет единицы, то команда 1 удваивает число, следовательно, получим следующее:

- 1: 14 => 28,
- 1: 28 => 56,
- 2: 56 => 57,
- 2: 57 => 58,
- 2: 58 => 59.

Ответ: 59.

Ответ: 59

6. Задание 6 № 3772

Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Python
<pre>DIM K, S AS INTEGER S = 3 K = 1 WHILE K &lt; 25   S = S + K   K = K + 2 WEND PRINT S</pre>	<pre>s = 3 k = 1 while k &lt; 25:   s += k   k += 2 print(s)</pre>
Паскаль	Алгоритмический язык
<pre>var k, s: integer; begin   s:=3;   k:=1;   while k &lt; 25 do begin     s:=s+k;     k:=k+2;   end;   write(s); end.</pre>	<pre>алг нач   цел k, s   s := 3   k := 1   нц пока k &lt; 25     s := s + k     k := k + 2   кц   ВЫВОД s кон</pre>
Си++	
<pre>#include &lt;iostream&gt; using namespace std; int main() {   int s, k;   s = 3, k = 1;   while (k &lt; 25) {     s = s + k;     k = k + 2;   }   cout &lt;&lt; s &lt;&lt; endl;   return 0; }</pre>	

**Решение.**

Цикл while выполняется до тех пор, пока истинно условие  $k < 25$ , т.к. по усл. while  $k < 25$  do begin и  $k:=k+2$ . То есть цикл будет выполнен 12 раз.

Аккуратно выпишем все  $s$  и  $k$ :

$s$  3 4 7 12 19 28 39 52 67 84 103 124 147

$k$  1 3 5 7 9 11 13 15 17 19 21 23 25

Таким образом, после завершения цикла получим  $s = 147$ .

Ответ: 147.

Ответ: 147

### 7. Задание 7 № 13736

Автоматическая фотокамера производит растровые изображения размером  $640 \times 480$  пикселей. При этом объём файла с изображением не может превышать 320 Кбайт, упаковка данных не производится. Какое максимальное количество цветов можно использовать в палитре?

**Решение.**

Объём растрового изображения находится как произведение количества пикселей в изображении на объём памяти  $x$ , необходимый для хранения цвета одного пиксела:  $640 \cdot 480 \cdot x \leq 320 \cdot 2^{10} \cdot 2^3$  бит, откуда  $x \leq 2^5 \cdot 2^{10} \cdot 2^3 / (2^{11} \cdot 15)$ , откуда находим  $x = 8$  бит. Значит, в изображении можно использовать не более  $2^8 = 256$  цветов.

Ответ: 256.

Ответ: 256

### 8. Задание 8 № 10473

Шифр кодового замка представляет собой последовательность из пяти символов, каждый из которых является цифрой от 1 до 4. Сколько различных вариантов шифра можно задать, если известно, что цифра 1 встречается ровно два раза, а каждая из других допустимых цифр может встречаться в шифре любое количество раз или не встречаться совсем?

**Решение.**

Количество способов поставить две 1 на пять позиций —  $5 \cdot \frac{4}{2} = 10$ .

После того, как определили позиции двух 1, на оставшиеся позиции можем поставить любое из трёх чисел, это можно сделать  $3^3 = 27$  способами.

Итого всего  $10 \cdot 27 = 270$  кодов.

Ответ: 270.

Ответ: 270

### 9. Задание 9 № 27521

Откройте файл электронной таблицы, содержащей вещественные числа — результаты ежечасного измерения температуры воздуха на протяжении трёх месяцев.

#### Задание 9

Сколько раз встречалась температура, которая равна минимальному значению?

**Решение.**

Для поиска минимального значения температуры воспользуемся формулой =МИН(B2:Y92). Минимальное значение температуры равно 8,1. Для счёта количества значений 8,1 воспользуемся формулой =СЧЁТЕСЛИ(B2:Y92; "8,1"). Такое значение встречается 1 раз.

Ответ: 1.

Ответ: 1

### 10. Задание 10 № 27583

С помощью текстового редактора определите, сколько раз, не считая сносок, встречается слово «Онегин» в тексте романа в стихах А. С. Пушкина «Евгений Онегин». Другие формы слова «Онегин», такие как «Онегина», «Онегиным» и т. д., учитывать не следует. В ответе укажите только число.

#### Задание 10

**Решение.**

Воспользуемся поисковыми средствами текстового редактора. В строке поиска введём слово «Онегин». Подсчитав общее количество результатов, получаем ответ — 12.

Ответ: 12.

Ответ: 12

11. Задание 11 № [5270](#)

При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 6 символов и содержащий только символы из 7-буквенного набора Н, О, Р, С, Т, У, Х. В базе данных для хранения сведений о каждом пользователе отведено одинаковое целое число байт, при этом для хранения сведений о 100 пользователях используется 1400 байт. Для каждого пользователя хранятся пароль и дополнительные сведения. Для хранения паролей используют посимвольное кодирование, все символы кодируются одинаковым и минимально возможным количеством бит. Сколько бит отведено для хранения дополнительных сведений о каждом пользователе?

**Решение.**

Для хранения сведений об одном пользователе используется  $\frac{1400}{100} = 14$  байт = 112 бит.

Согласно условию, в пароле могут быть использованы 7 букв. Известно, что с помощью  $N$  бит можно закодировать  $2^N$  различных вариантов. Поскольку  $2^2 < 7 < 2^3$ , то для записи каждой из 7 букв необходимо 3 бит. Для записи шести букв необходимо  $3 \cdot 6 = 18$  бит.

Тогда для хранения дополнительных сведений используется  $112 - 18 = 94$  бит.

Ответ: 94.

Ответ: 94

12. Задание 12 № **10388**

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** (111, 27)

преобразует строку 05111150 в строку 0527150.

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).

Ниже приведена программа для исполнителя Редактор.

НАЧАЛО

ПОКА **нашлось** (722) ИЛИ **нашлось** (557)

ЕСЛИ **нашлось** (722)

ТО **заменить** (722, 57)

ИНАЧЕ **заменить** (557, 72)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход этой программе подается строка, состоящая из 55 цифр; последняя цифра в строке — цифра 7, а остальные цифры — пятёрки. Какая строка получится в результате применения программы к этой строке? В ответе запишите полученную строку.

**Решение.**

Строка содержит 54 пятёрки и семерку.

По ходу работы программы строка будет меняться так: ...5557 → ...55572 → ...555722 → ...5557

При этом за 3 итерации цикла будет убираться  $2 + 2 - 1 = 3$  пятёрки.

Тогда после 17 раз по 3 итерации, то есть после 51 итераций, из строки будет убрано  $17 \cdot 3 = 51$  пятёрка и она примет вид 5557. После чего цикл отработает ещё один раз и заменит 5557 на 572.

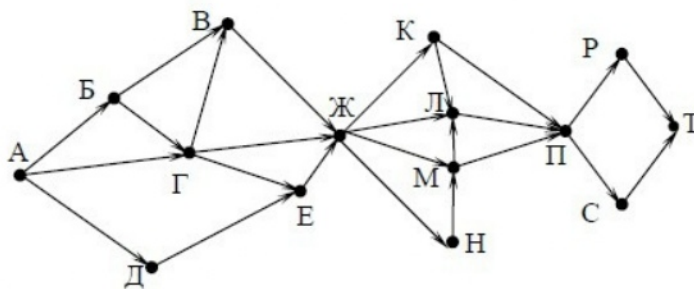
Ответ: 572



13. Задание 13 № 14776

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, К, Л, М, Н, П, Р, С, Т. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей из города А в город Т, проходящих через город Н?



**Решение.**

Количество путей до города Т равно сумме путей в каждый из тех городов, из которых есть дорога в Т. С помощью этого наблюдения посчитаем последовательно количество путей до каждого из городов, кроме тех, которые не проходят через город Л:

- A = 1
- B = A = 1
- D = A = 1
- Г = A + B = 1 + 1 = 2
- V = B + Г = 1 + 2 = 3
- E = Г + Д = 2 + 1 = 3
- Ж = V + Г + E = 3 + 2 + 3 = 8
- H = Ж = 8
- M = H = 8 (так как ищем пути проходящие через город Н)
- L = M = 8
- П = Л + М = 16
- Р = П = 16
- С = П = 16
- Т = 16 + 16 = 32

Ответ: 32.

Ответ: 32

14. Задание 14 № 9367

Значение арифметического выражения:  $9^8 + 3^5 - 9$  – записали в системе счисления с основанием 3. Сколько цифр «2» содержится в этой записи?

**Решение.**

Преобразуем выражение:

$$(3^2)^8 + 3^5 - 3^2$$

$$3^{16} + 3^5 - 3^2$$

$$3^{16} + 3^5 = 100\dots00100000$$

$$100\dots00100000 - 3^2 = 100\dots00022200$$

В полученном числе три двойки.

Ответ: 3

Ответ: 3

15. Задание 15 № [7675](#)

Элементами множества  $A$  являются натуральные числа. Известно, что выражение

$$(x \in \{2, 4, 6, 8, 10, 12\}) \rightarrow (((x \in \{3, 6, 9, 12, 15\}) \wedge \neg(x \in A)) \rightarrow \neg(x \in \{2, 4, 6, 8, 10, 12\}))$$

истинно (т. е. принимает значение 1) при любом значении переменной  $x$ . Определите наименьшее возможное значение суммы элементов множества  $A$ .

**Решение.**

Введем обозначения:

$$(x \in \{2, 4, 6, 8, 10, 12\}) \equiv P; (x \in \{3, 6, 9, 12, 15\}) \equiv Q; (x \in A) \equiv A.$$

Преобразовав, получаем:

$$P \rightarrow ((Q \wedge \neg A) \rightarrow \neg P) = P \rightarrow (\neg(Q \wedge \neg A) \vee \neg P) = \neg P \vee (\neg(Q \wedge \neg A) \vee \neg P) = \neg P \vee \neg Q \vee A.$$

Логическое ИЛИ истинно, если истинно хотя бы одно утверждение. Выражение  $\neg P \vee \neg Q$  истинно при всех значениях  $x$ , кроме значений 6 и 12. Следовательно, множество  $A$  должно содержать точки 6 и 12. То есть минимальный набор точек в множестве  $A \equiv \{6, 12\}$ . Сумма элементов множества  $A$  равна 18.

Ответ: 18.

Ответ: 18

16. Задание 16 № 10474

Ниже на пяти языках программирования записана рекурсивная функция (процедура)  $F$ .

Бейсик	Python
<pre>SUB F(n)   PRINT n,   IF n &gt; 2 THEN     F(n - 3)     F(n - 2)     F(n - 1)   END IF END SUB</pre>	<pre>def F(n):   print (n, end="")   if n &gt; 2:     F(n - 3)     F(n - 2)     F(n - 1)</pre>
Паскаль	Алгоритмический язык
<pre>procedure F(n: integer); begin   write(n);   if n &gt; 2 then   begin     F(n - 3);     F(n - 2);     F(n - 1)   end end;</pre>	<pre>алг F(цел n) нач   ВЫВОД n   если n &gt; 2 то     F(n - 3)     F(n - 2)     F(n - 1)   все кон</pre>
Си	
<pre>void F(int n){   cout &lt;&lt; n &lt;&lt; endl;   if (n &gt; 2) {     F(n - 3);     F(n - 2);     F(n - 1);   } }</pre>	

Что выведет программа при вызове  $F(4)$ ? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

**Решение.**

Рассмотрим структуру вызова функций, очередность.

```
F(4) {
  F(1)
  F(2)
  F(3) {
    F(0)
    F(1)
    F(2)
  }
}
```

Так как число печатается сразу при заходе в функцию, то порядок чисел будет совпадать с порядком вызовов.

Ответ: 4123012.

Ответ: 4123012

17. Задание 17 № [29665](#)

Определите количество принадлежащих отрезку  $[2 \cdot 10^{10}; 4 \cdot 10^{10}]$  натуральных чисел, которые делятся на 7 и на 100000 и при этом не делятся на 13, 29, 43 и 101, а также наименьшее из таких чисел. В ответе запишите два целых числа без пробелов и других дополнительных символов: сначала количество, затем наименьшее число.

Для выполнения этого задания можно написать программу или воспользоваться редактором электронных таблиц.

**Решение.**

Заметим, что, поскольку число должно делиться на 100000, можно поделить концы интервала на 100000. Это необходимо учесть при выводе ответа, умножим найденное минимальное число на 100000.

Приведём решение данной задачи на языке Паскаль:

```
var sum, min, i: longint;
begin
  min := 400001;
  sum := 0;
  for i := 200000 to 400000 do begin
    if i mod 7 = 0 then
      if i mod 13 <> 0 then
        if i mod 29 <> 0 then
          if i mod 43 <> 0 then
            if i mod 101 <> 0 then begin
              sum := sum + 1;
              if i < min then
                min := i;
            end;
          end;
        end;
      end;
    end;
  end;
  writeln(sum, 100000.0*min:12:0);
end.
```

Результат работы программы — 2462620000400000.

Ответ: 2462620000400000.

Ответ: 2462620000400000

18. Задание 18 № 33520

Дан квадрат  $15 \times 15$  клеток, в каждой клетке которого записано целое число. В левом верхнем углу квадрата стоит ладья. За один ход ладья может переместиться в пределах квадрата на любое количество клеток вправо или вниз (влево и вверх ладья ходить не может). Необходимо переместить ладью в правый нижний угол так, чтобы сумма чисел в клетках, в которых ладья останавливалась (включая начальную и конечную), была максимальной. В ответе запишите максимально возможную сумму.

Исходные данные записаны в электронной таблице.

Задание 18

*Пример входных данных (для таблицы размером  $4 \times 4$ ):*

-3	1	-3	-4
-4	-4	-2	2
6	1	2	-2
-6	7	6	-3

Для указанных входных данных ответом будет число 14 (ладья проходит через клетки с числами -3, 6, 1, 7, 6, -3).

**Решение.**

Скопируем число из ячейки A1 в ячейку P1. Поскольку ладья может ходить через неограниченное количество ячеек вниз и вправо, необходимо для каждой ячейки выбирать, из какого числа в строке до этой ячейки, и из какого числа в столбце выше этой ячейки должна сходить ладья, чтобы сумма ячеек при этом была максимальной. Для этого в ячейке Q1 запишем формулу `=МАКС($P$1:P1)+B1` и скопируем её во все ячейки диапазона R1:AD1. В ячейке P2 запишем формулу `=МАКС($P$1:P1)+A2` и скопируем её во все ячейки диапазона P3:P15. В ячейке Q2 запишем формулу `=МАКС(МАКС($P2:P2);МАКС(Q$1:Q1))+B2` и скопируем её во все ячейки диапазона Q2:AD15. Получим ответ — 323.

Ответ: 323.

Ответ: 323



19. Задание 19 № [27794](#)

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 один камень или  
 увеличить количество камней в куче в два раза .**

Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать  $(6, 9)$ . За один ход из позиции  $(6, 9)$  можно получить любую из четырёх позиций:  $(7, 9)$ ,  $(12, 9)$ ,  $(6, 10)$ ,  $(6, 18)$ . Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 50. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 50 или больше камней.

В начальный момент в первой куче было 8 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 ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение  $S$ , когда такая ситуация возможна.

**Решение.**

Минимальное значение:  $S = 11$ . Петя может получить позицию  $(8, 22)$ , в которой Ваня может выиграть ходом  $(8, 44)$ . При меньших значениях  $S$  ни при каком ходе Пети Ваня не сможет выиграть первым ходом.

Ответ: 11.

Ответ: 11

20. Задание 20 № [27795](#)

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 один камень или  
 увеличить количество камней в куче в два раза.**

Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать  $(6, 9)$ . За один ход из позиции  $(6, 9)$  можно получить любую из четырёх позиций:  $(7, 9)$ ,  $(12, 9)$ ,  $(6, 10)$ ,  $(6, 18)$ . Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 50. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 50 или больше камней.

В начальный момент в первой куче было 8 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 ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Найдите два таких значения  $S$ , при которых 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.

Найденные значения запишите в ответе в порядке возрастания без разделительных знаков.

**Решение.**

Возможные значения  $S$ : 16, 20. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при  $S = 16$  Петя может получить позицию  $(16, 16)$ , а при  $S = 20$  — позицию  $(9, 20)$ .

В первом случае после хода Вани возникнет одна из позиций  $(17, 16)$ ,  $(32, 16)$ ,  $(16, 17)$ ,  $(16, 32)$ , во втором случае — одна из позиций  $(10, 20)$ ,  $(18, 20)$ ,  $(9, 21)$ ,  $(9, 40)$ . В любой из перечисленных позиций Петя может выиграть, удвоив количество камней в большей куче.

Таким образом, ответ — 1620.

Ответ: 1620.

Ответ: 1620

21. Задание 21 № [27796](#)

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 один камень или  
 увеличить количество камней в куче в два раза.**

Например, пусть в одной куче 6 камней, а в другой 9 камней; такую позицию мы будем обозначать (6, 9). За один ход из позиции (6, 9) можно получить любую из четырёх позиций: (7, 9), (12, 9), (6, 10), (6, 18). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 50. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 50 или больше камней.

В начальный момент в первой куче было 8 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 описании выигрышной стратегии не следует включать ходы играющего по ней игрока, которые не являются для него безусловно выигрышными, то есть не гарантируют выигрыш независимо от игры противника.

Найдите минимальное 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.

**Решение.**

Возможное значение  $S$ : 19. После первого хода Пети возможны позиции (9, 19), (16, 19), (8, 20), (8, 38). В позициях (16, 19) и (8, 38) Ваня может выиграть первым ходом, удвоив количество камней в любой куче. Из позиций (9, 19) и (8, 20) Ваня может получить позицию (9, 20), в этом случае после хода Пети возникнет одна из позиций (10, 20), (18, 20), (9, 21), (9, 40). В любой из перечисленных позиций Ваня может выиграть, удвоив количество камней в большей куче.

Таким образом, ответ — 19.

Ответ: 19.

Ответ: 19

22. Задание 22 № [3209](#)

Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

Бейсик	Python
DIM X, L, M AS	
INTEGER	
INPUT X	
L = 0	x = int(input())
M = 0	L = 0
WHILE X > 0	M = 0
L = L + 1	while x > 0:
IF X MOD 2 = 0 THEN	L += 1
M = M + (X MOD 10)	if x % 2 == 0:
2	M = M + (x % 10) // 2
ENDIF	x = x // 10
X = X \ 10	print(L)

WEND PRINT L PRINT M	print(M)
<b>Паскаль</b>	<b>Алгоритмический язык</b>
<pre>var x, L, M: integer; begin   readln(x);   L := 0;   M := 0;   while x &gt; 0 do   begin     L := L + 1;     if x mod 2 = 0 then       M := M + (x mod 10) div 2;     x := x div 10;   end;   writeln(L);   writeln(M); end.</pre>	<pre>алг нач   цел x, L, M   ввод x   L := 0   M := 0   нц пока x &gt; 0     L := L + 1     если mod(x,2) = 0       то         M := M + div(mod(x,10), 2)     все     x := div(x,10)   кц   вывод L, нс, M кон</pre>
<b>Си++</b>	
<pre>#include &lt;iostream&gt; using namespace std; int main() {   int x, L, M;   cin &gt;&gt; x;   L = 0;   M = 0;   while (x &gt; 0){     L = L + 1;     if(x % 2 == 0){       M = M + (x % 10) / 2;     }     x = x / 10;   }   cout &lt;&lt; L &lt;&lt; endl &lt;&lt; M endl; }</pre>	



**Решение.**

Рассмотрим цикл, число шагов которого зависит от изменения переменной  $x$ :

```
while x > 0 do begin
...
x:= x div 10;
end;
```

Т. к. оператор `div` оставляет только целую часть от деления, то при делении на 10 это равносильно отсечению последней цифры.

Из приведенного цикла видно, что на каждом шаге от десятичной записи  $x$  отсекается последняя цифра до тех пор, пока все цифры не будут отсечены, то есть  $x$  не станет равно 0; поэтому цикл выполняется столько раз, сколько цифр в десятичной записи введенного числа, при этом число  $L$  столько же раз увеличивается на 1. Следовательно, конечное значение  $L$  совпадает с числом цифр в  $x$ . Для того, чтобы  $L$  стало  $L=3$ ,  $x$  должно быть **трёхзначным**.

Теперь рассмотрим оператор изменения  $M$ :

```
if x mod 2 = 0 then
M:= M + (x mod 10) div 2;
end;
```

Оператор `mod` оставляет только остаток от деления, при делении на 10 это последняя цифра  $x$ .

Условие  $x \bmod 2 = 0$  означает следующее: чтобы  $M$  увеличилось, число  $x$  должно быть чётным.

Предположим, исходное  $x$  нечётное, тогда на первом шаге  $M = 0$ .

Если на втором шаге  $x$  также нечётное (вторая цифра исходного числа нечётная), то  $M = 0$ , причём каким бы ни было значение  $x$  на третьем шаге, мы не сможем получить  $M = 7$ , поскольку остаток от деления чётного числа на 10 не превосходит 8, а  $8 / 2 = 4$ , следовательно, **вторая цифра исходного  $x$  чётная**.

Тогда первая цифра может принимать значения 2, 4, 6, 8, но мы ищем наибольшее  $x$ , поэтому сделаем первую цифру, равной 9, тогда наше предположение не удовлетворяет условию задачи, и последняя цифра исходного числа обязана быть чётной, т.е. **исходное  $x$  чётно**.

$7 = 4 + 3$ , чему соответствуют цифры 8 и 6. Теперь, располагая цифры по убыванию, находим наибольшее возможное  $x$ :  $x = 986$ .

Ответ: 986.

Ответ: 986



23. Задание 23 № **13418**

Исполнитель НечетМ преобразует число на экране. У исполнителя НечетМ две команды, которым присвоены номера:

1. прибавь 1
2. сделай нечётное

Первая из этих команд увеличивает число  $x$  на экране на 1, вторая переводит число  $x$  в число  $2x+1$ . Например, вторая команда переводит число 10 в число 21. Программа для исполнителя НечетМ – это последовательность команд. Сколько существует таких программ, которые число 1 преобразуют в число 27, причём траектория вычислений не содержит число 26? Траектория вычислений программы – это последовательность результатов выполнения всех команд программы. Например, для программы 121 при исходном числе 7 траектория будет состоять из чисел 8, 17, 18.

**Решение.**

Используем метод динамического программирования. Заведём массив  $dp$ , где  $dp[i]$  — кол-во способов получить число  $i$  с помощью таких команд.

База динамики:

$$dp[1]=1;$$

Формула перехода:

$$dp[i]=dp[i-1] \text{ - если } i \text{ - четное.}$$

$$dp[i]=dp[i-1] + dp[(i-1)/2] \text{ - если } i \text{ нечетное.}$$

Но при этом, если  $i-1 = 26$  или  $(i-1)/2 = 26$ , то оно не учитывается. Можно заметить, что для числа 27 будет формула  $dp[27]=dp[26] + dp[13]$ , а поскольку  $dp[26]$  не считается, то ответ совпадает с  $dp[13]$ .

Посчитаем  $dp[13]$  (далее будут приведены значения в ячейках  $dp$  от 1 до 13):  
 1 1 2 2 3 3 5 5 7 7  
 10 10 13.

Ответ: 13.

Ответ: 13

24. Задание 24 № [27697](#)

Текстовый файл состоит не более чем из  $10^6$  символов  $L$ ,  $D$  и  $R$ . Определите длину самой длинной последовательности, состоящей из символов  $D$ . Хотя бы один символ  $D$  находится в последовательности.

Для выполнения этого задания следует написать программу. Ниже приведён файл, который необходимо обработать с помощью данного алгоритма.

[Задание 24](#)

**Решение.**

Для решения данной задачи будем посимвольно считывать текстовый файл. Объявим переменные:  $maxLen$  — максимальная длина последовательности,  $curLen$  — временное хранение длины последовательности,  $i$  — переменная для перебора всех символов,  $s$  — строка для работы с символами из файла. Алгоритм будет сравнивать значение текущего символа со значением предыдущего и, если символы не будут различаться и будут являться буквой  $D$ , то значение счетчика будет увеличиваться на 1.

**Приведём решение данной задачи на языке Pascal.**

```
var maxLen, curLen, i: integer;
s: string;
begin
assign(input, '24.txt');
readln(s);
maxLen := 1;
curLen := 1;
for i:=2 to Length(s) do
if (s[i] = s[i-1]) and (s[i] = 'D') then begin
curLen := curLen + 1;
if curLen > maxLen then
maxLen := curLen;
end
else
curLen := 1;
writeln(maxLen);
end.
```

В результате работы данного алгоритма при вводе данных из файла в условии получаем ответ — 11.

Ответ: 11.

Ответ: 11

25. Задание 25 № [28122](#)

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку  $[489421; 489440]$ , числа, имеющие ровно четыре различных натуральных делителя. Для каждого найденного числа запишите эти четыре делителя в четыре соседних столбца на экране с новой строки. Делители в строке должны следовать в порядке возрастания.

Например, в диапазоне  $[12; 14]$  ровно четыре различных натуральных делителя имеет число 14, поэтому для этого диапазона вывод на экране должна содержать следующие значения:

1 2 7 14

Ответ:


Решение.

Решим задачу перебором. Будем проверять количество делителей каждого числа из диапазона, если их количество равно четырём — записываем их в двумерный массив  $d$ . После этого выводим эти делители на экран в новой строке.

Приведём решение на языке Pascal.

```
var
x, numDel, i, j: longint;
d: array[1..4] of longint;
begin
for i := 489421 to 489440 do begin
numDel := 0;
for j := 1 to i do begin
if i mod j = 0 then begin
numDel := numDel + 1;
if numDel > 4 then break;
d[numDel] := j;
end;
end;
if numDel = 4 then writeln(d[1], ' ', d[2], ' ', d[3], ' ', d[4]);
end;
end.
```

В результате работы программа должна вывести следующее:

```
1 19 25759 489421
1 2 244711 489422
1 13 37649 489437
```

Ответ: 1 & 19 & 25759 & 489421 & 1 & 2 & 244711 & 489422 & 1 & 13 & 37649 & 489437

26. Задание 26 № [27883](#)

Системный администратор раз в неделю создаёт архив пользовательских файлов. Однако объём диска, куда он помещает архив, может быть меньше, чем суммарный объём архивируемых файлов. Известно, какой объём занимает файл каждого пользователя.

По заданной информации об объёме файлов пользователей и свободном объёме на архивном диске определите максимальное число пользователей, чьи файлы можно сохранить в архиве, а также максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

**Входные данные.**

Задание 26

В первой строке входного файла находятся два числа:  $S$  — размер свободного места на диске (натуральное число, не превышающее 10 000) и  $N$  — количество пользователей (натуральное число, не превышающее 3000). В следующих  $N$  строках находятся значения объёмов файлов каждого пользователя (все числа натуральные, не превышающие 100), каждое в отдельной строке.

Запишите в ответе два числа: сначала наибольшее число пользователей, чьи файлы могут быть помещены в архив, затем максимальный размер имеющегося файла, который может быть сохранён в архиве, при условии, что сохранены файлы максимально возможного числа пользователей.

Пример входного файла:

100 4

80

30

50

40

При таких исходных данных можно сохранить файлы максимум двух пользователей. Возможные объёмы этих двух файлов 30 и 40, 30 и 50 или 40 и 50. Наибольший объём файла из перечисленных пар — 50, поэтому ответ для приведённого примера:

2 50

Ответ:

--	--

**Решение.**

Сначала считаем в массив данные из файла. После этого отсортируем массив в порядке возрастания. Таким образом, последовательно складывая элементы массива с начала и сравнивая сумму с размером свободного места на диске получим максимальное количество пользователей, чьи файлы могут поместиться на диске. Далее, вычитая из найденной суммы наибольший файл в текущей последовательности, будем пробовать прибавлять файлы с большим весом. Если такой файл будет найден, то заменяем значение наибольшего файла, который возможно поместить на диск.

**Приведём решение на языке Pascal.**

```
var
i, j, t: integer;
a: array [1..3000] of integer;
s: integer;
n: integer;
sum: integer;
maxi: integer;
f: text;
begin
assign(f,'C:\27883.txt');
reset(f);
readln(f, s, n);
for i := 1 to n do readln(f, a[i]);
for i := 1 to n do
for j := i + 1 to n do
if a[i] > a[j] then begin
t := a[i];
a[i] := a[j];
a[j] := t;
end;
sum := 0;
maxi := 1;
for i := 1 to n do
if sum + a[i] <= s then begin
sum := sum + a[i];
maxi := i;
end;
t := a[maxi];
for i := maxi to n do
if ((sum - t) + a[i]) <= s then begin
sum := sum - t + a[i];
t := a[i];
end;
writeln(maxi, ', ', t);
end.
```

Ответ: 149 7.

*Примечание.* Путь к файлу необходимо указать согласно расположению файла на Вашем компьютере.

Ответ: 149&7

**27. Задание 27 № 28129**

На вход программы поступает последовательность из  $N$  натуральных чисел. Рассматриваются все пары различных элементов последовательности, у которых различные остатки от деления на  $d = 160$  и хотя бы одно из чисел делится на  $p = 7$ . Среди таких пар, необходимо найти и вывести пару с максимальной суммой элементов.



**Входные данные.**

[Файл А](#)

[Файл В](#)

В первой строке входных данных задаётся количество чисел  $N$  ( $1 \leq N \leq 1000$ ). В каждой из последующих  $N$  строк записано одно натуральное число, не превышающее 10000. В качестве результата программа должна напечатать элементы искомой пары. Если среди найденных пар максимальную сумму имеют несколько, то можно напечатать любую из них. Если таких пар нет, то вывести два нуля.

Пример организации исходных данных во входном файле:

4  
 168  
 7  
 320  
 328

Пример выходных данных для приведённого выше примера входных данных:

168 320

В ответе укажите четыре числа: сначала значение искомой пары для файла *A* (два числа через пробел по возрастанию), затем для файла *B* (два числа через пробел по возрастанию).

Ответ:

**Решение.**

Отметим:

$m71$  самое большое число кратное 7;

$m72$  второе по величине число кратное 7 и остаток от деления на 160 не равен остатку от деления  $m71$  на 160;

$m1$  самое большое число не кратное 7;

$m2$  второе по величине число не кратное 7 и остаток от деления на 160 не равен остатку от деления  $m71$  на 160.

**Приведём решение задачи на языке Pascal.**

```
var
n, i, m71, m72, m1, m2, max1, max2, x: integer;
f: text;
begin

m71:=0;
m72:=0;
m1:=0;
m2:=0;
max1:=0;
max2:=0;
assign(f,'28129_A.txt');
reset(f);
readln(f, N);
while not eof(f) do begin
readln(f, x);
if (x mod 7 = 0) and (x mod 160 = m71 mod 160) and (x > m71) then
m71 := x
else if (x mod 7 = 0) and (x mod 160 <> m71 mod 160) and (x > m71) then
begin
m72 := m71;
m71 := x;
end
else if (x mod 7 = 0) and (x mod 160 <> m71 mod 160) and (x > m72) then
m72 := x
```

```

else if (x mod 7 <> 0) and (x mod 160 = m1 mod 160) and (x > m1) then
m1 := x
else if (x mod 7 <> 0) and (x mod 160 <> m1 mod 160) and (x > m1) then
begin
m2 := m1;
m1 := x;
end
else if (x mod 7 <> 0) and (x mod 160 <> m1 mod 160) and (x > m2) then
m2 := x;
end;
if (m71 = 0) and (m72 = 0) then
writeln(0, ' ', 0)
else if (m72 = 0) and (m2 = 0) and (m71 mod 160 = m1 mod 160) then
writeln(0, ' ', 0)
else
begin
if (m71+m72)>(max1+max2) then
begin
max1 := m71;
max2 := m72;
end;
if ((m71+m1)>(max1+max2)) and (m71 mod 160 <> m1 mod 160) then
begin
max1 := m71;
max2 := m1;
end;
if ((m71+m2)>(max1+max2)) and (m71 mod 160 <> m2 mod 160) then
begin
max1 := m71;
max2 := m2;
end;
if (((m72 + m1) > (max1 + max2)) and ((m72 mod 160) <> (m1 mod 160))) then
begin
max1 := m72;
max2 := m1;
end;
writeln(max1, ' ', max2);
end;
end.

```

**Приведём решение Ивана Корниенко на языке Pascal.**

```

var
s, n, x, x1, k7, n7, t: integer;
a: array[0..159, 0..1] of integer;
f: text;
begin
for var j := 0 to 159 do
for var l := 0 to 1 do
a[j][l] := 0;
s := 0;
assign(f, 'C:\28129_A.txt');
reset(f);
readln(f, n);
readln(f, x);
for var i := 2 to n do
begin
t := 1;

```

```

if x mod 7 = 0 then
t := 0;
if x >= a[x mod 160][t] then
a[x mod 160][t] := x;
readln(f, x);
for var k := 0 to 159 do
for var r := 0 to 1 do
if ((x + a[k][r]) > s) and (x mod 160 <> k) and
((x * a[k][r]) mod 7 = 0) then
begin
s := x + a[k][r];
x1 := x;
end;
end;
if s = 0 then
writeln('00')
else
if (s - x1) < x1 then writeln(s - x1, ' ', x1)
else writeln(x1, ' ', s - x1)
end.

```

В результате работы данного алгоритма при вводе данных из файла А ответ — 728 977, из файла В — 9982 9992.

*Примечание.* Путь к файлу необходимо указать согласно расположению файла на Вашем компьютере.

Ответ: 728 977&9982 9992