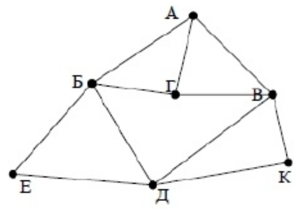
Вариант № 9169593

1. **Задание 1 №** 15098

На рисунке схема дорог изображена в виде графа, в таблице содержатся сведения о длине этих дорог в километрах.

	П1	П2	П3	П4	П5	П6	П7
П1		20		14		19	18
П2	20		14		16		15
П3		14		18	15		
П4	14		18		17	14	
П5		16	15	17			
П6	19			14			
П7	18	15					



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Известно, что длина кратчайшего пути из пункта А в пункт К не превышает 30 километров. Определите длину кратчайшего пути из пункта Г в пункт Е. В ответе укажите целое число — длину пути в километрах.

Решение.

1. Д — единственная вершина степени 4, у которой нет дорог к населённым пунктам А и Г, значит, Д соответствует пункту П1.

2. Можно заметить, что есть два населённых пункта с тремя дорогами, три населённых пункта с четырьмя дорогами, и два населённых пункта с двумя дорогами. Из этого можно сделать вывод, что П3 и П5 это либо А, либо Г, населённые пункты П1, П2 и П4 это Д, Б или В, а населённые пункты П6 и П7 это Е или К.

3. Предположим, что П3 это А, а П7 это К. Из этого можно сделать вывод: П5 это Г, П6 это Е, П2 это В, а П4 это Б. После этого находим кратчайший путь между вершинами А и К, который будет равен 29, что соответствует условию задания. Значит, предположение оказалось верным.

4. Далее находим кратчайший путь между вершинами Г и Е, который будет равен 31.

Ответ: 31.
Ответ: 31

2. Задание 2 № 9752

Логическая функция F задаётся выражением:

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

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

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

Перем. 1	Перем. 2	Перем. 3	Функция
???	???	???	F
0	0	0	1
1	0	0	1
1	0	1	1

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. Пусть задано выражение $x \rightarrow y$, зависящее от двух переменных x и y, и таблица истинности:

Перем. 1	Перем. 2	Функция
???	???	F
0	0	1
0	1	0
1	0	1
1	1	1

Тогда 1-му столбцу соответствует переменная y, а 2-му столбцу соответствует переменная x. В ответе нужно написать: yx.

Решение.

Рассмотрим данное выражение. Оно равно единице в трех случаях: $(\neg x \wedge y \wedge z) = 1$, $(\neg x \wedge \neg y \wedge z) = 1$ или $(\neg x \wedge \neg y \wedge \neg z) = 1$. Каждое из этих равенств выполняется только при одном наборе переменных. Первое: $x = 0$, $y = 1$, $z = 1$. Второе: $x = 0$, $y = 0$, $z = 1$. Третье: $x = y = z = 0$. Так, из второго значения функции видим, что переменная 1 — z. А из третьего, что переменная 2 — x, тогда переменная 3 — y.

Ответ: zxy.

Приведем другое решение (Владимир Юрьевич Ламок).

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

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

Сопоставим эти наборы с приведенным в задании фрагментом таблицы истинности.

Во всех наборах переменная x принимает значение 0, следовательно, переменной x соответствует второй столбец таблицы.

Переменная y принимает единичное значение только в одном наборе, следовательно, ей соответствует третий столбец таблицы, тогда первый столбец соответствует переменной z.

Для составления таблицы истинности можно воспользоваться программой на языке Python:

```
for x in range(2):
  for y in range(2):
    for z in range(2):
      if (not(x) and y and z) or (not(x) and not(y) and z) or (not(x) and not (y) and not(z))==1:
        print(x, y, z)
```

Ответ: zxy

3. Задание 3 № 15099

Даны фрагменты двух таблиц из базы данных. Каждая строка таблицы 2 содержит информацию о ребёнке и об одном из его родителей. Информация представлена значением поля ID в соответствующей строке таблицы 1. На основании имеющихся данных определите, у скольких детей отец старше матери более чем на 2 года. При вычислении ответа учитывайте только информацию из приведённых фрагментов таблиц.

Таблица 1

ID	Фамилия И.О.	Пол	Год рождения
127	Грищенко А.В.	М	1936
148	Грищенко Д.И.	М	1998
182	Грищенко Е.П.	Ж	1940
212	Грищенко И.А.	М	1970
243	Грищенко Н.Н.	Ж	1976
254	Клейн А.Б.	М	1984
314	Клейн Е.А.	Ж	2009
412	Клейн М.А.	Ж	2011
543	Панько О.А.	Ж	1948
544	Петров В.И.	М	1961
545	Петров О.В.	М	1991
750	Петрова А.Е.	Ж	1962
830	Седых А.Н.	Ж	1980
849	Седых Н.Н.	М	1943

Таблица 2

ID Родителя	ID Ребенка
127	212
182	212
212	148
243	148
254	314
254	412
543	243
543	830
544	545
750	545
830	314
830	412
849	243
849	830

Решение.

Используя данные таблиц, вычислим возраст родителей (первым идёт ID ребёнка (детей), вторым идут данные родителей).

ID 212: мать Грищенко Е.П., ID 182, отец Грищенко А.В, ID 127 — подходит.

ID 148: мать Грищенко Н.Н., ID 243, отец Грищенко И.А., ID 212 — подходит.

ID 314 и ID 412: мать Седых А.Н., ID 830, отец Клейн А.Б, ID 254 — не подходит, поскольку отец младше матери.

ID 243 и ID 830: мать Панько О.А., ID 543, отец Седых Н.Н., ID 849 — подходят (двое детей).

ID 545: мать Петрова А.Е., ID 750, отец Петров В.И., ID 544 — не подходит, поскольку отец старше матери меньше, чем на 2 года.

Таким образом, у четырёх детей отец старше матери более чем на 2 года.

Ответ: 4.

Примечание.

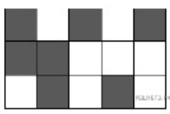
Заметим, что если один человек старше другого, что его год рождения меньше.

Ответ: 4

4. Задание 4 № 3682

Черно-белое растровое изображение кодируется построчно, начиная с левого верхнего угла и заканчивая в правом нижнем углу. При кодировании 1 обозначает черный цвет, а 0 — белый.

Закодируйте таким образом изображение и запишите результат в восьмеричной системе счисления.



Решение.

Код первой строки: 10101.

Код второй строки: 11000.

Код третьей строки: 01010.

Запишем коды по порядку в одну строку: 101011100001010. Теперь разобьём это представление на тройки справа налево и переведём полученный набор чисел в десятичный код (восьмеричное предствление совпадает с десятичным при разбиении тройками).

101 011 100 001 010 — 53412.

Ответ: 53412

5. Задание 5 № 10495

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 10000 преобразуется в запись 100001;
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 такое наименьшее число N, для которого результат работы алгоритма больше 97. В ответе это число запишите в десятичной системе счисления.

Решение.

Если изначально сумма разрядов была чётная, то в конец запишется 00, что эквивалентно $N \to N \cdot 4$.

Если же сумма была нечётная, то запишется 10, что эквивалентно $N \to N \cdot 4 + 2$.

В обоих случаях число получается чётным.

Посмотрим на чётные числа, превосходящие 97.

$98_{10} = 1100010_2$ — на конце 10, но сумма остальных разрядов чётна, не подходит ни под один из случаев.

$100_{10} = 1100100_2$ — на конце 00, но сумма остальных разрядов нечётна, не подходит.

$102_{10} = 1100110_2$ — на конце 10, а сумма остальных разрядов нечётна. Число подходит под второй случай, значит, число, из которого оно было получено, равно $\frac{102-2}{4} = 25$.

Ответ: 25

6. Задание 6 № 3243

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

Бейсик	Python

DIM K, S AS INTEGER
S = 0
K = 0
WHILE S < 80
  S = S + 2*K
  K = K + 4
   WEND
   PRINT S
	s = 0
	k = 0
	while s < 80:
	  s += 2*k
	  k += 4
	print(s)

Паскаль	**Алгоритмический язык**

var k, s: integer;
begin
   s:=0;
   k:=0;
   while s < 80 do begin
     s:=s+2*k;
     k:=k+4;
   end;
   write(s);
end.
	алг
	нач
	  цел k, s
	  s := 0
	  k := 0
	  нц пока s < 80
	    s := s + 2*k
	    k := k + 4
	  кц
	  вывод s
	кон

Си++

```
#include <iostream>
using namespace std;
int main() {
    int s, k;
    s = 0, k = 0;
    while (s < 80) {
        s = s + 2*k;
        k = k + 4;
    }
    cout << s << endl;
    return 0;
}
```

Решение.

Цикл while выполняется до тех пор, пока истинно условие s < 80, т. е. переменная s определяет, сколько раз выполнится цикл.

Аккуратно выпишем все s и k:

s 0 0 8 24 48 80
k 0 4 8 12 16 20

(Помните, что условие s < 80 проверяется только после k:=k+4, поэтому действие s:=s+2*k последний раз выполнится для k=16)

Следовательно ответ 80.

Ответ: 80

7. Задание 7 № 4585

Текстовый документ хранился в 8-битной кодировке КОИ-8. Этот документ был преобразован в 16-битную кодировку Unicode, при этом размер памяти, необходимой для хранения документа увеличился на 4 Кбайт. При этом хранится только последовательность кодов символов. Укажите, сколько символов в документе. В ответе запишите только число.

Решение.

Обозначим количество символов в документе за x.

Тогда объем информации в кодировке КОИ-8: $8x$бит = x байт

Объем информации в 16-битной кодировке Unicode: $16x$ бит = $2x$байт.

Размер памяти увеличился на $2x - x$ байт = 4*1024 байт .

Откуда $x = 4096$.

Ответ: 4096 символов.

Ответ: 4096

8. Задание 8 № 4794

В коробке лежат 64 цветных карандаша. Сообщение о том, что достали белый карандаш, несет 4 бита информации. Сколько белых карандашей было в коробке?

Решение.

Формула Шеннона: $x = log_2\left(\frac{1}{p}\right)$, где x — количество информации в сообщении о событии P, p — вероятность события P.

Вероятность достать из коробки белый карандаш $p = \frac{y}{64}$.

Воспользовавшись формулой Шеннона, получаем, что $4 = x = log_2\left(\frac{64}{y}\right) = 6 - log_2 y$.

Следовательно, $y = 4$.

Ответ: 4

9. Задание 9 № 27522

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

Задание 9

Сколько раз встречалась температура, выше округленного до десятых среднего арифметического значения всех чисел в таблице?

Решение.

Для поиска среднего арифметического значения температуры воспользуемся формулой **=СРЗНАЧ(B2:Y92)**. Среднее арифметическое значение температуры равно 23,9. Теперь с помощью формулы **=СЧЁТЕСЛИ(B2:Y92; ">23,9")** найдём количество измерений, которые выше среднего арифметического значения — 1192.

Ответ: 1192.

Ответ: 1192

10. Задание 10 № 27589

С помощью текстового редактора определите, сколько раз, не считая сносок, встречается слово «мы» или «Мы» в тексте романа в стихах А. С. Пушкина «Евгений Онегин». Другие формы слова «мы» учитывать не следует. В ответе укажите только число.

Задание 10

Решение.

Воспользуемся поисковыми средствами текстового редактора. В строке поиска введем слово «мы». Подсчитав общее количество результатов, получаем ответ — 8.

Ответ: 8.

Ответ: 8

11. Задание 11 № 7332

Автомобильный номер состоит из нескольких букв (количество букв одинаковое во всех номерах), за которыми следуют 4 цифры. При этом используются 10 цифр и только 4 буквы: А, В, Т, О. Нужно иметь не менее 1 000 000 различных номеров. Какое наименьшее количество букв должно быть в автомобильном номере?

Решение.

В алфавите состоящем из N символов N^M слов длиной M символов. Пусть L — длина части номера, состоящей из букв. Тогда, при помощи цифр и букв мы можем закодировать $4^L \cdot 10^4$ номеров. Значит, для кодирования 1 000 000 номеров нужно минимально $\log_4(10^6/10^4) = \log_4 100$ букв. Следовательно, минимально нужно использовать четыре буквы.

Ответ: 4

12. Задание 12 № 4584

Исполнитель РОБОТ умеет перемещаться по прямоугольному лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними по сторонам клетками может стоять стена.

Система команд исполнителя РОБОТ содержит восемь команд. Четыре команды - это команды перемещения:

вверх	**вниз**	**влево**	**вправо**

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если на пути РОБОТа окажется стена, он разрушится.

Четыре команды проверяют отсутствие стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	**снизу свободно**	**слева свободно**	**справа свободно**

Цикл

ПОКА <условие>

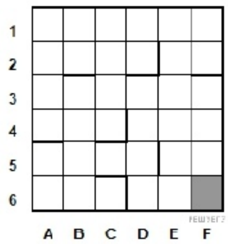
последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкциях ПОКА условие может содержать команды проверки, а также слова И, ИЛИ, НЕ.

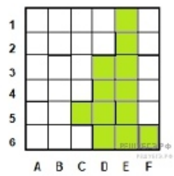
Схема лабиринта:



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

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

Решение.



При данной программе РОБОТ поступает следующим образом: сперва РОБОТ проверяет свободна ли клетка справа или снизу от него, если это так, то РОБОТ переходит к первому действию внутри цикла. В этом цикле пока снизу клетки в которой находится РОБОТ нет стены он продолжает двигаться вниз. Как только это условие перестанет выполняться он переходит ко второму действию цикла. Второе действие, заключается в следующем: РОБОТ передвигается на одну клетку вправо. После чего возвращается к началу внешнего цикла. У робота есть возможность разбиться: например, стартовав из любой клетки столбца F робот разобьётся.

Проверив все клетки по выведенному нами правилу движения РОБОТА выясняем, что число клеток, удовлетворяющих условию задачи, равно 12.

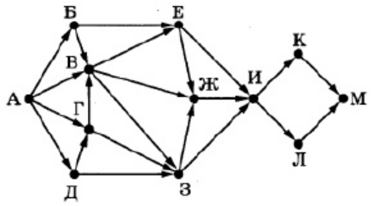
Ответ: 12.

Ответ: 12

13. Задание 13 № 10478

На рисунке представлена схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, З, И, К, Л, М. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой.

Сколько существует различных путей из города А в город М, проходящих через город Ж, но не проходящих через город К?



Решение.

Количество путей до города Х = количество путей добраться в любой из тех городов, из которых есть дорога в Х.

При этом если путь должен не проходить через какой-то город, нужно просто не учитывать этот город при подсчёте сумм. А если город наоборот обязательно должен лежать на пути, тогда для городов, в которые из нужного города идут дороги, в суммах нужно брать только этот город.

С помощью этого наблюдения посчитаем последовательно количество путей до каждого из городов:

А = 1
Б = А = 1
Д = А = 1
Г = А + Д = 1 + 1 = 2
В = А + Б + Г = 1 + 1 + 2 = 4
Е = Б + В = 1 + 4 = 5
З = В + Г + Д = 4 + 2 + 1 = 7
Ж = В + Е + З = 4 + 5 + 7 = 16
И = Ж = 16
К = И = 16
Л = И = 16

Таким образом, путей, проходящих через город Ж, но не проходящих через город К: М = Л = 16.

Ответ: 16.
Ответ: 16

14. Задание 14 № 2308

Укажите через запятую в порядке возрастания все десятичные числа, не превосходящие 30, запись которых в системе счисления с основанием 5 начинается на 3?

Решение.

Сначала определим запись числа 30 в пятеричной системе. $30_{10} = 110_5$. Выпишем числа, не большие 110_5, запись которых в пятеричной системе начинается на 3: 3, 30, 31, 32, 33, 34.

Переведем их в десятичную систему счисления. $3_5 = 3_{10}$, $30_5 = 15_{10}$, $31_5 = 16_{10}$, $32_5 = 17_{10}$, $33_5 = 18_{10}$, $34_5 = 19_{10}$.

Ответ: 3,15,16,17,18,19

15. Задание 15 № 14779

Сколько существует целых значений числа A, при которых формула

$$((x < 5) \to (x^2 < A)) \wedge ((y^2 \le A) \to (y \le 5))$$

тождественно истинна при любых целых неотрицательных x и y?

Решение.

Раскрывая импликацию по правилу A → B = ¬A + B, заменяя логическую сумму совокупностью, а логическое произведение системой соотношений, определим значения параметра A, при котором система совокупностей

$$\begin{cases} \left[\begin{array}{l} x \ge 5, \\ x^2 < A, \end{array}\right. \\ \left[\begin{array}{l} y^2 > A, \\ y \le 5 \end{array}\right. \end{cases}$$

будет иметь решениями для любых целых неотрицательных чисел.

Заметим, что переменные не связаны между собой уравнением или неравенством, поэтому необходимо и достаточно, чтобы решениями первой совокупности были все неотрицательные x, а решениями второй совокупности были все неотрицательные y.

Решениями неравенства $x \ge 5$ являются числа 5, 6, 7 ... Чтобы совокупность выполнялась для всех целых неотрицательных чисел, числа 0, 1, 2, ... 4 должны быть решениями неравенства $x^2 < A$. Значит, $A > 16$.

Аналогично, решениями неравенства $y \le 5$ являются числа 0, 1, ... 5. Следовательно, числа 6, 7, 8 ... должны быть решениями неравенства $y^2 > A$. Поэтому $A < 36$.

Тем самым, $16 < A < 36$. Искомое количество целых значение параметра равно 19.

Ответ: 19.
Ответ: 19

16. Задание 16 № 13738

Ниже на пяти языках программирования записан рекурсивный алгоритм *F*.

Бейсик	Python

DECLARE SUB F(n)
SUB F(n)
  IF n > 0 THEN
    PRINT n
    F(n - 3)
    F(n \ 3)
  END IF
END SUB
	def F(n):
	  if n > 0:
	    print(n)
	    F(n - 3)
	    F(n // 3)

Паскаль	**Алгоритмический язык**

procedure F(n: integer);
begin
  if n > 0 then begin
    writeln(n);
    F(n - 3);
    F(n div 3)
  end
end;
	алг F(цел n)
	нач
	  если n > 0 то
	    вывод n, нс
	    F(n - 3)
	    F(div(n, 3))
	  все
	кон

Си++	

void F(int n) {
  if (n > 0) {
    std: :cout << n;
    F(n - 3);
    F(n / 3);
  }
}	

Запишите подряд без пробелов и разделителей все числа, которые будут напечатаны на экране при выполнении вызова F(9). Числа должны быть записаны в том же порядке, в котором они выводятся на экран.

Решение.

Промоделируем работу алгоритма, не выписывая F с аргументом меньше трёх.

```
F(9)
  F(6)
    F(3)
      F(0)
      F(1)
        F(-2)
    F(2)
      F(-1)
      F(0)
  F(3)
    F(0)
    F(1)
```

Таким образом, при выполнении вызова F(9) будут напечатаны числа 9631231.

Ответ: 9631231.

Ответ: 9631231

17. Задание 17 № 27626

Рассматривается множество целых чисел, принадлежащих числовому отрезку [1721; 4322], которые делятся на 3 и 11 и не делятся на 5, 9, 13, 22. Найдите количество таких чисел и максимальное из них. В ответе запишите два целых числа без пробелов и других дополнительных символов: сначала количество, затем максимальное число.

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

Решение.

Приведём решение данной задачи на языке Паскаль:

```
var count, max, i: integer;
begin
max := 0;
count := 0;
for i := 1721 to 4322 do begin
if (i mod 3 = 0) and (i mod 11 = 0) then
if i mod 5 <> 0 then
if i mod 9 <> 0 then
if i mod 13 <> 0 then
if i mod 22 <> 0 then begin
count := count + 1;
if i > max then
max := i;
end;
end;
writeln(count, max);
end.
```

Ответ: 194191.

Ответ: 194191

18. Задание 18 № 29666

Дана последовательность вещественных чисел. Из неё необходимо выбрать несколько подряд идущих чисел так, чтобы каждое следующее число было меньше предыдущего. Какую максимальную сумму могут иметь выбранные числа?

В ответе запишите только целую часть максимально возможной суммы. Исходная последовательность записана в виде одного столбца электронной таблицы.

Задание 18

Пример входных данных:

5,2
3,1
1,2
2,3
7,1
3,3

Для указанных входных максимально возможная сумма равна 10,4, в ответе надо записать число 10.

Решение.

Скопируем число из ячейки A1 в ячейку B1. Далее будем сравнивать текущее число с предыдущим и, если оно текущее число будет меньше предыдущего, суммировать текущее число с предыдущим числом в столбце B. Запишем в ячейку B2 формулу **=ЕСЛИ(A2<A1;A2+B1;A2)** и скопируем эту формулу во все ячейки диапазона **B3:B500**. Таким образом, все искомые суммы последовательностей будут найдены. Теперь в ячейке C1 запишем формулу **=МАКС(B1:B500)**, получив значение 358,76. Таким образом, ответ — 358.

Ответ: 358.

Ответ: 358

19. Задание 19 № 27754

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 четыре 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), (24, 9), (6, 10), (6, 36). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 61 или больше камней.

В начальный момент в первой куче было 3 камня, во второй куче — S камней, $1 \le S \le 57$.

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.

Заметим, что игра должна закончиться в 2 хода. Минимальное значение количества камней в обеих кучах, при котором игра заканчивается — 61. Эта ситуация возможна, например, когда в первой куче 3 камня, а во второй — 58. Значит, чтобы Ваня мог выиграть своим первым ходом, количество камней во второй куче после первого хода Пети должно быть ≥ 15. Это возможно при значении $S = 4$. При таком минимальном значении S Ваня выиграет своим первым ходом после неудачного хода Пети.

Ответ:4.
Ответ: 4

20. Задание 20 № 27755

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 четыре 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), (24, 9), (6, 10), (6, 36). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 61 или больше камней.

В начальный момент в первой куче было 3 камня, во второй куче — S камней, $1 \le S \le 57$.

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: 12, 14. В этих случаях Петя, очевидно, не может выиграть первым ходом. Однако при $S = 12$ Петя может получить позицию (12, 12), а при $S = 14$ — позицию (4, 14).

В первом случае после хода Вани возникнет одна из позиций (13, 12), (48, 12), (12, 13), (12, 48), во втором случае — одна из позиций (5, 14), (16, 14), (4, 15), (4, 56). В любой из перечисленных позиций Петя может выиграть, умножив количество камней во второй куче.

Таким образом, ответ — 1214.

Ответ: 1214.
Ответ: 1214

21. Задание 21 № 27756

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 четыре 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), (24, 9), (6, 10), (6, 36). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 61. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 61 или больше камней.

В начальный момент в первой куче было 3 камня, во второй куче — S камней, $1 \leq S \leq 57$.

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 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.

Минимальное значение S: 13. После первого хода Пети возможны позиции (4, 13), (12, 13), (3, 14), (3, 52). В позициях (12, 13) и (3, 52) Ваня может выиграть первым ходом, умножив количество камней в любой куче. Из позиций (4, 13) и (3, 14) Ваня может получить позицию (4, 14), в этом случае возникнет одна из позиций (5, 14), (16, 14), (4, 15), (4, 56). В любой из перечисленных позиций Ваня может выиграть, умножив количество камней во второй куче.

Таким образом, ответ — 13.

Ответ: 13.

Ответ: 13

22. Задание 22 № 6781

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 2.

Бейсик

```
DIM X, A, B AS INTEGER
INPUT X
A=0: B=0
WHILE X > 0
   A = A+1
   IF B < (X MOD 8) THEN
      B = X MOD 8
   END IF
   X = X \ 8
WEND
PRINT A
PRINT B
```

Паскаль

```
var x, a, b: integer;
   begin
   readln(x);
   a:=0; b:=0;
   while x>0 do
      begin
         a:=a + 1;
         if b < (x mod 8)
         then
            b:=x mod 8;
         x:=x div 8;
      end;
   writeln(a); write(b);
end.
```

Си++

```
#include <iostream>
using namespace std;
int main()
{
   int x, a, b;
   cin >> x;
   a=0; b=0;
   while (x>0){
      a = a+1;
      if (b < (x%8)){
         b = x%8;
      }
      x = x/8;
   }
   cout << a << endl << b << endl;
}
```

Алгоритмический

```
алг
нач
цел x, a, b
ввод x
a:=0; b:=0
нц пока x>0
   a:=a+1
   если b < mod(x,8)
      то
         b:=mod(x,8)
   все
   x:=div(x,8)
кц
вывод a, нс, b
кон
```

Python

```
x = int(input())
a = 0
b = 0
while x > 0:
   a += 1
   if (b < (x % 8)):
      b =x % 8
   x //= 8
print(a)
print(b)
```

Решение.

Значение в переменной *a* после выполнения цикла равно количеству выполненных циклов. Поскольку требуется, чтобы программа напечатала сначала число 3, цикл должен выполниться три раза. Оператор div оставляет только целую часть от деления, следовательно, искомое число должно два раза делиться на 8 так, чтобы целая часть результата была положительной. Следовательно, это число должно быть не меньше числа 64.

В переменную *b* записывается остаток от деления числа на 8. По условию требуется, чтобы после выполнения цикла переменная *b* имела значение 2, т. е. максимальный остаток от деления на 8 в цикле должен быть равен 2.

Выполним программу для всех чисел, не меньших чем 64. Первое число, которое удовлетворит условию и будет наименьшим. Поскольку программа выводит целые числа и никаких других операторов к числу, кроме операторов div и mod, не применяется, будем рассматривать только целые числа.

При вводе числа 64 программа выведет числа 3 и 1. При вводе числа 65 программа выведет числа 3 и 1. При вводе числа 66 программа выведет числа 3 и 2. Следовательно, ответ 66.

Ответ: 66.

Ответ: 66

23. Задание 23 № 3527

У исполнителя Множик есть две команды:

1. умножь на 8,
2. подели на 2.

Первая из них увеличивает число на экране в 8 раз, вторая – уменьшает его в 2 раза.

Программа для Множика – это последовательность команд. Сколько различных чисел можно получить из числа 512 с помощью программы, которая содержит ровно 8 команд?

Решение.

От перестановок множителей произведение не меняется, поэтому, подсчитав количество возможных программ, найдём количество разных чисел. Запишем все программы в виде набора команд, с точностью до перестановки:

1. 1 1 1 1 1 1 1 1,
2. 2 1 1 1 1 1 1 1,
3. 2 2 1 1 1 1 1 1,
4. 2 2 2 1 1 1 1 1,
5. 2 2 2 2 1 1 1 1,
6. 2 2 2 2 2 1 1 1,
7. 2 2 2 2 2 2 1 1,
8. 2 2 2 2 2 2 2 1,
9. 2 2 2 2 2 2 2 2.

Всего получили 9 различных программ, дающие 9 различных чисел.

Ответ: 9.

Ответ: 9

24. Задание 24 № 27696

Текстовый файл состоит не более чем из 10^6 символов *L*, *D* и *R*. Определите длину самой длинной последовательности, состоящей из символов *L*. Хотя бы один символ *L* 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 L, 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] = 'L') 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 — 7.

Ответ: 7.

Ответ: 7

25. Задание 25 № 27855

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку [95632; 95700], числа, имеющие ровно шесть различных чётных натуральных делителей (при этом количество нечётных делителей может быть любым). Для каждого найденного числа запишите эти шесть делителей в шесть соседних столбцов на экране с новой строки. Делители в строке должны следовать в порядке возрастания.

Например, в диапазоне [2; 48] ровно шесть чётных различных натуральных делителей имеют числа 24, 36 и 40, поэтому для этого диапазона вывод на экране должна содержать следующие значения:

2 4 6 8 12 24
2 4 6 12 18 36
2 4 8 10 20 40

Ответ:

Решение.

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

Приведём решение на языке Pascal.

```
var
numDel, i, j: longint;
d2: array[1..6] of longint;
begin
for i := 95632 to 95700 do begin
numDel := 0;
for j := 1 to i do begin
if (i mod j = 0) and (j mod 2 = 0) then begin
numDel := numDel + 1;
if numDel > 6 then break;
d2[numDel] := j;
end;
end;
if numDel = 6 then writeln(d2[1], ' ', d2[2], ' ', d2[3], ' ', d2[4], ' ', d2[5], ' ', d2[6]);
end;
end.
```

В результате работы программа должна вывести следующее:

2 10 50 3826 19130 95650
2 26 338 566 7358 95654
2 4 8 23918 47836 95672

Ответ:
2&10&50&3826&19130&95650&2&26&338&566&7358&95654&2&4&8&23918&47836&95672

26. Задание 26 № 28138

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 2000). 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..2000] of integer;
s: integer;
n: integer;
sum: integer;
maxi: integer;
f: text;
begin
assign(f,'C:\28138.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: 509 31.

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

Ответ: 509&31

27. Задание 27 № 33772

Набор данных состоит из нечётного количества пар натуральных чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы чётность суммы выбранных чисел совпадала с чётностью большинства выбранных чисел и при этом сумма выбранных чисел была как можно меньше. Определите минимальную сумму, которую можно получить при таком выборе. Гарантируется, что удовлетворяющий условиям выбор возможен.

Входные данные.

Файл A
Файл B

Первая строка входного файла содержит число N — общее количество пар в наборе. Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример входного файла:

5
15 8
5 11

6 3
7 2
9 14

Для указанных данных надо выбрать числа 8, 5, 3, 2 и 9. Большинство из них нечётны, сумма выбранных чисел равна 27 и тоже нечётна. В ответе надо записать число 27.

Вам даны два входных файла (*A* и *B*), каждый из которых имеет описанную выше структуру. В ответе укажите два числа: сначала значение искомой суммы для файла *A*, затем для файла *B*.

Предупреждение: для обработки файла *B* не следует использовать переборный алгоритм, вычисляющий сумму для всех возможных вариантов, поскольку написанная по такому алгоритму программа будет выполняться слишком долго.

Ответ:

Решение.

Последовательно считывая данные из файла, будем прибавлять к сумме значение минимального числа в паре, при этом, если число чётное, будем увеличивать значение переменной count0 на единицу, если нечётное — увеличивать значение переменной count1 на единицу. Поскольку может возникнуть ситуация, когда, например, получившаяся сумма будет чётной, а количество чётных чисел будет меньше количества нечётных чисел и будет отличаться от количества нечётных чисел на единицу, будем находить две минимальных разницы для ситуации, когда будет убираться два чётных числа (переменные dif3 и dif4), и две минимальных разницы для ситуации, когда будет убираться два нечётных числа (переменные dif1 и dif2).

Приведём решение задачи на языке Pascal.

```
var
x, y, count0, count1: integer;
n: integer;
sum: integer;
dif1, dif2, dif3, dif4: integer;
f: text;
begin
    assign(f,'C:\27-B.txt');
    reset(f);
    readln(f, n);
    sum := 0;
    dif1 := 20001;
    dif2 := 20001;
    dif3 := 20001;
    dif4 := 20001;
    count0 := 0;
    count1 := 0;
    while not eof(f) do begin
        readln(f, x, y);
        if x < y then begin
            sum := sum + x;
            if x mod 2 = 0 then count0 := count0 + 1
            else count1 := count1 + 1;
            if x mod 2 <> y mod 2 then begin
                if (y - x < dif1) and (y mod 2 <> 0) then begin
                    dif2 := dif1;
                    dif1 := y - x;
                end
                else if (y - x < dif2) and (y mod 2 <> 0) then
                    dif2 := y - x;
                if (y - x < dif3) and (y mod 2 = 0) then begin
                    dif4 := dif3;
                    dif3 := y - x;
                end
                else if (y - x < dif4) and (y mod 2 = 0) then
                    dif4 := y - x;
            end;
        end
        else begin
            if y mod 2 = 0 then count0 := count0 + 1
            else count1 := count1 + 1;
            sum := sum + y;
            if x mod 2 <> y mod 2 then begin
                if (x - y < dif1) and (x mod 2 <> 0) then begin
                    dif2 := dif1;
                    dif1 := x - y;
                end
                else if (x - y < dif2) and (x mod 2 <> 0) then
                    dif2 := x - y;
                if (x - y < dif3) and (x mod 2 = 0) then begin
```

```
                    dif4 := dif3;
                    dif3 := x - y;
                end
                else if (x - y < dif4) and (x mod 2 = 0) then
                    dif4 := x - y;
            end;
        end;
    end;
    if (count1 > count0) then begin
        if sum mod 2 = 1 then
            writeln(sum)
        else if dif1 <= dif3 then
            writeln(sum + dif1)
        else if (count1 - count0) <> 1 then
            writeln(sum + dif3)
        else if (dif3 + dif4) < dif1 then
            writeln(sum + dif3 + dif4)
        else writeln(sum + dif1)
    end
    else begin
        if sum mod 2 = 0 then
            writeln(sum)
        else if dif3 <= dif1 then
            writeln(sum + dif3)
        else if (count0 - count1) <> 1 then
            writeln(sum + dif1)
        else if (dif1 + dif2) < dif3 then
            writeln(sum + dif1 + dif2)
        else writeln(sum + dif3)
    end;
end.
```

В результате работы данного алгоритма при вводе данных из файла A ответ — 61772, из файла B — 18484085.

Примечание. Путь к файлу необходимо указать согласно расположению файла на Вашем компьютере.
Ответ: 61772&18484085