

**Единый государственный экзамен
по ИНФОРМАТИКЕ и ИКТ**

Инструкция по выполнению работы

Экзаменационная работа состоит из 27 заданий с кратким ответом, выполняемых с помощью компьютера.

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

Экзаменационная работа выполняется с помощью специализированного программного обеспечения, предназначенного для проведения экзамена в компьютерной форме. При выполнении заданий Вам будут доступны на протяжении всего экзамена текстовый редактор, редактор электронных таблиц, системы программирования. Расположение указанного программного обеспечения на компьютере и каталог для создания электронных файлов при выполнении заданий Вам укажет организатор в аудитории.

На протяжении сдачи экзамена доступ к сети Интернет запрещён.

При выполнении заданий можно пользоваться черновиком. **Записи в черновике не учитываются при оценивании работы.**

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

Желаем успеха!

В экзаменационных заданиях используются следующие соглашения.

1. Обозначения для логических связей (операций):

- a) отрицание (инверсия, логическое НЕ) обозначается \neg (например, $\neg A$);
- b) конъюнкция (логическое умножение, логическое И) обозначается \wedge (например, $A \wedge B$) либо $\&$ (например, $A \& B$);
- c) дизъюнкция (логическое сложение, логическое ИЛИ) обозначается \vee (например, $A \vee B$) либо $|$ (например, $A | B$);
- d) следование (импликация) обозначается \rightarrow (например, $A \rightarrow B$);
- e) тождество обозначается \equiv (например, $A \equiv B$). Выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо они оба истинны, либо они оба ложны);
- f) символ 1 используется для обозначения истины (истинного высказывания); символ 0 – для обозначения лжи (ложного высказывания).

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

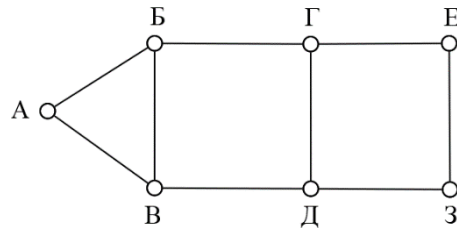
3. Приоритеты логических операций: инверсия (отрицание), конъюнкция (логическое умножение), дизъюнкция (логическое сложение), импликация (следование), тождество. Таким образом, $\neg A \wedge B \vee C \wedge D$ означает то же, что и $((\neg A) \wedge B) \vee (C \wedge D)$.

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

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

1 На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах). Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. В таблице в левом столбце указаны номера пунктов, откуда совершается движение, в первой строке – куда. Сумму длин дорог из пункта Г в пункт Е и из пункта Д в З.

	П1	П2	П3	П4	П5	П6	П7
П1						12	7
П2					10	11	9
П3				5	6	3	
П4			5		15		
П5		10	6	15			
П6	12	11	3				
П7	7	9					



Решение 2, аналитическое.

Так как граф симметричный, невозможно однозначно идентифицировать дороги ГЕ и ДЗ относительно друг друга. Однако можно пару дорог, которые однозначно совпадают с искомыми двумя дорогами.

Для этого найдем два пункта Е и З. Это пункты с двумя дорогами, которые имеют общую дорогу.

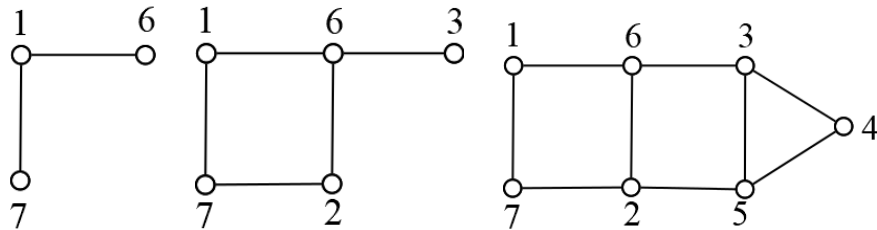
Пункты с двумя дорогами – П1, П4 и П7. П1 и П7 имеют общую дорогу. Следовательно, остальные две дороги в парах – искомые дороги.

П1-П6 и П7-П2 – длиной 12 и 9 соответственно.

Ответ: 21

Решение 1, графическое.

Построим по таблице граф. Начнем с любого пункта, из которого исходит две дороги.



Исходя из построенного графа можно заключить, что искомые дороги это 1-6 и 7-2.

Ответ: 21

2 Логическая функция F задаётся выражением $(\neg a \wedge \neg b) \vee (b \equiv c) \vee d$. На рисунке приведён частично заполненный фрагмент таблицы истинности функции F , содержащий **неповторяющиеся строки**. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных a, b, c, d .

?	?	?	?	F
		1		0
1	0		1	0
0	0	1	1	0

В ответе напишите буквы a, b, c, d в том порядке, в котором идут соответствующие им столбцы. Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		$\neg x \vee y$
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму столбцу – переменная x . В ответе следует написать yx .

Решение 1, аналитическое.

$d = 0$, иначе выражение будет истинным. Сразу же можно определить столбец, в котором находятся значения переменной d – столбец 2.

Найдем значения переменных a, b, c , при которых первое и второе подвыражения ложны.

a	b
0	1
1	0
1	1

b	c
0	1
1	0

→

a	b	c
0	1	0
1	0	1
1	1	0

Следовательно, c – первый столбец (единственно возможный с одной единицей), b – третий (единственный возможный для инвертированных значений c), a – четвертый по остаточному принципу.

Ответ: cdba

Решение 2, полупрограммное.

Напишем программу, которая генерирует таблицу истинности.

Python
<pre>from itertools import * for a, b, c, d in product([0, 1], repeat=4): if (not a and not b) or (b == c) or d: print(a, b, c, d)</pre>
Pascal
<pre>var a, b, c, d: boolean; begin for a := False to True do for b := False to True do for c := False to True do for d := False to True do if (not a and not b) or (b=c) or d then writeln(ord(a), ord(b), ord(c), ord(d)) end.</pre>
C++
<pre>#include <iostream> using namespace std; int main(){ for(int a = 0; a <= 1; a++) for(int b = 0; b <= 1; b++) for(int c = 0; c <= 1; c++) for(int d = 0; d <= 1; d++) if((!a and !b) (b == c) d) cout << a << ' ' << b << ' ' << c << ' ' << d; }</pre>

После чего проведем анализ, аналогичный решению 1.

Ответ: cdba

3

В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведенных данных определите, сколько людей родилось в том же городе, что и один из их внуков или одна из их внучек?

Таблица 1

ID	Фамилия И.О.	Пол	Город
127	Височко Г.Г.	М	Брянск
148	Январин З.И.	М	Тула
182	Феврина М.А.	Ж	Тула
212	Мартшейн А.В.	М	Курск
243	Апрелько Е.С.	Ж	Москва
254	Май Н.А.	М	Курск
314	Июнина П.Е.	Ж	Тула
412	Июлон Л.Е.	Ж	Ижевск
543	Августович Т.О.	Ж	Тула
544	Сентябин О.С.	М	Курск
545	Окто Е.Н.	М	Брянск
750	Нояркина Б.Р.	Ж	Тула
830	Декабрь З.М.	Ж	Курск
849	Годин Ф.Ф.	М	Тула

Таблица 2

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

Решение

Найдем идентификаторы, которые есть как в первом, так и во втором столбце таблицы 2. Это люди, которые одновременно являются родителями и детьми. Следовательно, их родители являются бабушкой и дедушкой.

Родители-дети	Внуки	Бабушки-дедушки
182	412 Ижевск	127 Брянск
412	543 Тула	212 Курск 182 Тула
212	412 Ижевск	314 Тула 148 Тула
243	849 Тула	314 Тула 148 Тула
849	544 Курск 545 Брянск	243 Москва
314	212 Курск 243 Москва	254 Курск

Ответ: 4

4

Для кодирования некоторой последовательности используют следующую кодировочную таблицу.

А	00
Б	1001
В	1010
Г	110
Д	0101

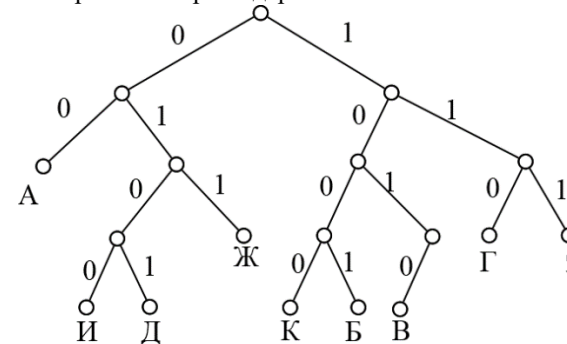
Е	
Ж	011
З	111
И	0100
К	1000

Укажите код минимальной длины для буквы Е, такой что будет соблюдаться условие Фано. Если таких кодов несколько, укажите код с минимальным числовым значением.

Примечание: условие Фано выполняется, когда ни одно кодовое слово не является началом другого кодового слова.

Решение

Построим бинарное дерево.



На дереве видно, что мы можем использовать только одну свободную ветку – 1011.

Ответ: 1011

5 Автомат обрабатывает натуральное число N по следующему алгоритму:

1. Строится двоичная запись числа N .
2. Если N четное, то в конец полученной записи (справа) дописывается 0, в начало – 1; если N – нечётное в конец и начало дописывается по две единицы.
3. Результат переводится в десятичную систему и выводится на экран.

Пример. Дано число $N = 13$. Алгоритм работает следующим образом:

1. Двоичная запись числа N : 1101.
2. Число нечетное, следовательно по две единицы по краям – 11110111.
3. На экран выводится число 247.

Укажите наименьшее число, большее 52, которое может являться результатом работы автомата.

Решение:

Переведем 52_{10} в двоичную систему счисления, получим 110100_2 .

Найдем два числа – минимальное четное, из которого можно получить двоичную последовательность такой же длины, и минимальное нечетное.

Для четного:

$\pm 10100 \rightarrow 1010$ из такого числа получается 110100, что равно исходному, нам необходимо найти БОльшее значение.

Следующее четное значение – 1100, при его преобразовании получим $111000_2 = 56_{10}$.

Для нечетного:

$\pm 10100 \rightarrow 01 \rightarrow 1$. Однако из такого числа получится 11111, что меньше заданного. Поэтому найдем минимальное двузначное нечетное двоичное число.

$11 \rightarrow 111111_2 = 63_{10}$

Минимальное из найденных значений – 56.

Ответ: 56

6 Какое максимальное значение переменной s , подаваемое на вход программе, для которого в результате работы программы на экран будет выведено значение 64? Для Вашего удобства программа представлена на нескольких языках программирования.

С++	Паскаль
<pre>#include <iostream> using namespace std; int main() { int n, s; n = 1024; cin >> s; while(s >= 5) { s = s - 5; n = n / 2; } cout << n; return 0; }</pre>	<pre>var n, s: integer; begin n := 1024; readln(s); while s >= 5 do begin s := s - 5; n := n div 2 end; write(n) end.</pre>
Python	Алгоритмический язык
<pre>n = 1024 s = int(input()) while s >= 5: s = s - 5 n = n // 2 print(n)</pre>	<pre>алг нач цел s, n n := 1024 ввод s нц пока s >= 5 s := s - 5 n := div(n, 2) кц вывод n кон</pre>

Решение:

Заметим, что после работы алгоритма число 1024 преобразуется в 64. На каждой итерации значение уменьшается вдвое. Следовательно, выполнится 4 итерации цикла.

$$\begin{cases} s - 3 \cdot 5 \geq 5 \\ s - 4 \cdot 5 < 5 \end{cases} \quad \begin{cases} s \geq 20 \\ s < 25 \end{cases} \quad 20 \leq s < 25$$

Ответ: 24

7 Изображение размером 315x3072 пикселей сохраняется в памяти компьютера. Для его хранения выделяется не более 735 Кбайт без учёта заголовка файла. Все пиксели кодируются одинаковым количеством бит и записываются в файл один за другим. Какое максимальное количество цветов может использоваться для хранения такого изображения?

В ответе запишите только число.

Решение:

Для вычисления максимального количества цветов необходимо определить, сколько бит можно выделить для хранения одного пикселя. Поэтому поделим объем изображения (в битах) на количество пикселей.

$$\frac{735 \cdot 2^{10} \cdot 2^3}{315 \cdot 3072} = \frac{245 \cdot 2^3}{315} = \frac{56}{9} = 6\frac{2}{9}$$

Значит максимальное ЦЕЛОЕ количество бит, которое можно использовать для кодирования цвета пикселя равно 6.

Следовательно, количество цветов $2^6 = 64$.

Ответ: 64

8 Ученица составляет 5-буквенные слова из букв ГЕПАРД. При этом в каждом слове ровно одна буква Г, слово не может начинаться на букву А и заканчиваться буквой Е. Какое количество слов может составить ученица?

Решение 1, аналитическое:

Имеем 5 возможных конфигураций строки.

Г {ЕПАРД} {ЕПАРД} {ЕПАРД} {ПАРД}
 {ЕПРД} Г {ЕПАРД} {ЕПАРД} {ПАРД}
 {ЕПРД} {ЕПАРД} Г {ЕПАРД} {ПАРД}
 {ЕПРД} {ЕПАРД} {ЕПАРД} Г {ПАРД}
 {ЕПРД} {ЕПАРД} {ЕПАРД} {ЕПАРД} Г

$1 \cdot 5 \cdot 5 \cdot 5 \cdot 4 = 500$
$4 \cdot 1 \cdot 5 \cdot 5 \cdot 4 = 400$
$4 \cdot 5 \cdot 1 \cdot 5 \cdot 4 = 400$
$4 \cdot 5 \cdot 5 \cdot 4 \cdot 4 = 400$
$4 \cdot 5 \cdot 5 \cdot 5 \cdot 1 = 500$

Ответ: 2200

Решение 2, программное.

```
Python
from itertools import *

c = 0
for w in product('ГЕПАРД', repeat=5):
    if w.count('Г')==1 and w[0]!='А' and w[-1]!='Е':
        c += 1
print(c)
```

Ответ: 2200



Задание выполняется с использованием прилагаемых файлов.

9

Откройте файл электронной таблицы, содержащей результаты метеорологических наблюдений. Найдите разницу между максимальной температурой в июле и минимальной температурой в октябре.

В ответе запишите только целую часть полученного результата.

Решение:

Найдем необходимые показатели

А	В	А	В	С
	Максимальная температура за день, С		Максимальная температура за день, С	Минимальная температура за день, С
01.07.2020	26,2	29.09.2020	23,5	14,5
02.07.2020	24,2	30.09.2020	15,2	3,6
03.07.2020	24,3	01.10.2020	-9,6	-11,7
04.07.2020	26,7	02.10.2020	13,9	-3,9
05.07.2020	34,9	03.10.2020	0,9	-5
06.07.2020	34,2	04.10.2020	19,2	14,4
07.07.2020	30,8	05.10.2020	19,1	1,6
08.07.2020	22,7	06.10.2020	14,4	10,6
09.07.2020	30,9	07.10.2020	-1,9	-12,5
10.07.2020	27,5	08.10.2020	14,2	3,2
11.07.2020	25,7	09.10.2020	-9,1	-12,5
12.07.2020	25,8	10.10.2020	9	2,5
13.07.2020	23,3	11.10.2020	17,1	-2,5
14.07.2020	20,9	12.10.2020	9,5	-0,8
15.07.2020	25,9	13.10.2020	-1,4	-10,1
16.07.2020	31,8	14.10.2020	13,5	-12,6
17.07.2020	31,7	15.10.2020	-2,9	-10,1
18.07.2020	25,1	16.10.2020	18,8	0,2
19.07.2020	33,3	17.10.2020	-0,5	-14,4
20.07.2020	25,8	18.10.2020	1,5	-4,4
21.07.2020	24,2	19.10.2020	-5,2	-8,1
22.07.2020	26,6	20.10.2020	12,1	0,7
23.07.2020	34,1	21.10.2020	5,8	-13,3
24.07.2020	24,8	22.10.2020	-4	-11
25.07.2020	23,1	23.10.2020	0,5	-4
26.07.2020	31,3	24.10.2020	2,4	-9,4
27.07.2020	23,9	25.10.2020	9,6	-4,6
28.07.2020	24	26.10.2020	16,1	7
29.07.2020	21,8	27.10.2020	7,9	-0,6
30.07.2020	31,4	28.10.2020	2	-14,7
31.07.2020	25,7	29.10.2020	14,3	12,4
		30.10.2020	15,1	2,5
		31.10.2020	14,5	-12

Напишем формулу для нахождения ответа:
 =ЦЕЛОЕ(МАКС(В2:В32)-МИН(С94:С124))

Ответ: 49



Задание выполняется с использованием прилагаемых файлов.

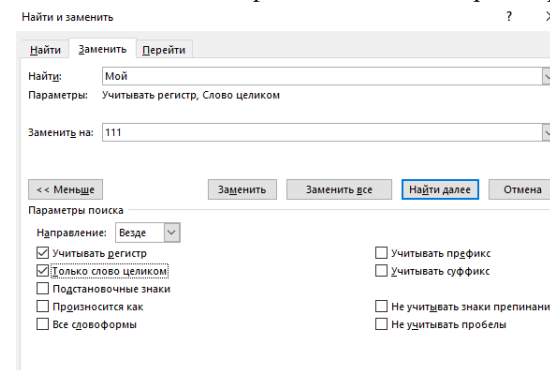
10

В файле дан текст поэмы Н.А.Некрасова «Кому на Руси жить хорошо». Сколько раз в тексте поэмы встречается слово «Мой» написанное с прописной буквы. Другие формы слова «Мой», такие как «Мои», «Моего» и пр. не учитывать.

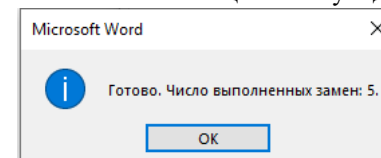
В ответе укажите только число.

Решение:

Вызовем меню замены и настроим замены по заданию. Слово «Мой», флажок «Только слово целиком», флажок «Учитывая регистр».



Нажмем «Заменить все». Во всплывающем окне увидим количество замен.



Ответ: 5

11

При регистрации в компьютерной системе каждому пользователю выдается идентификатор из 101 символа, каждый из которых может быть десятичной цифрой или одним из 4090 символов из специального набора. Каждый символ кодируется с помощью одинакового и минимального количества бит. Идентификатор же записывается в памяти с помощью минимально возможного целого количества байт.

Сколько килобайт потребуется для хранения идентификаторов 2048 пользователей?

Решение:

Для хранения одного из 4100 символов ($4090 + 10$) необходимо выделить 13 бит, $2^{12} \leq 4100 < 2^{13}$

Значит для хранения идентификатора нужно
 $13 \cdot 101 = 1313$ бит = 164.125 байт
 Минимальное целое количество байт – 165.

Для определения количества килобайт, необходимого для хранения идентификаторов 2048 пользователей нужно

$$\frac{165 \cdot 2048}{1024} = 330 \text{ Кбайт}$$

Ответ: 330

12

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

заменить (v, w)

нашлось (v)

Первая команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Если цепочки v в строке нет, эта команда не изменяет строку. Вторая команда проверяет, встречается ли цепочка v в строке исполнителя Редактор.

На выполнение Редактору дана следующая программа:

```
ПОКА нашлось (111) или нашлось (88888)
  ЕСЛИ нашлось (111)
    ТО заменить (111, 88)
  ИНАЧЕ заменить (88888, 8)
КОНЕЦ ПОКА
```

На вход программе подана строка из 100 единиц.

Какая строка будет получена исполнителем после выполнения данного алгоритма?

Решение 1, аналитическое.

Заметим, что по приоритету сначала заменяется 111 на 88, и только если в строке нет 111, то 88888 заменяется на 8.

Значит 99 единиц сначала заменятся на $33 \cdot 2$ восьмерки. Получим строку, вида 888...8881.

Операция замены 88888 на 8 удаляет из строки 4 восьмерки. $66 = 16 \cdot 4 + 2$. Следовательно, после преобразования получим строку 881.

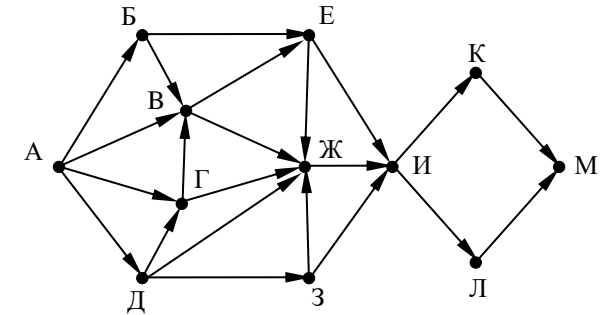
Ответ: 881

Решение 2, программное.

Python <pre>s = '1'*100 while '111' in s or '88888' in s: if '111' in s: s = s.replace('111', '88', 1) else: s = s.replace('88888', '8', 1) print(s)</pre>
PascalABC.net <pre>var s: string; begin s := '1'*100; while ('111' in s) or ('88888' in s) do if '111' in s then s := s.Replace('111', '88', 1) else s := s.Replace('88888', '8', 1); writeln(s); end.</pre>
C++ <pre>#include <iostream> #include <string> using namespace std; int main(){ string s = string(100, '1'); while(s.find("111") != string::npos s.find("88888") != string::npos) if(s.find("111") != string::npos) s.replace(s.find("111"), 3, "88"); else s.replace(s.find("88888"), 5, "8"); cout << s; }</pre>

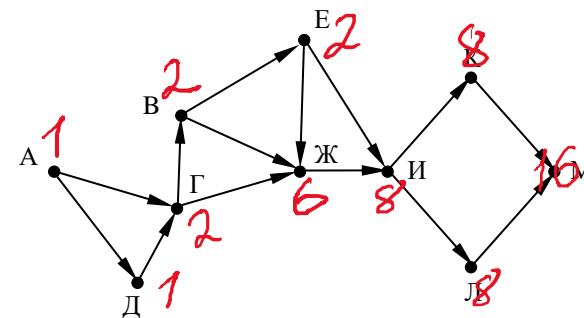
Ответ: 881

13 На рисунке – 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 Г?



Решение:

Построим все маршруты в Г и из Г.



Ответ: 16

14 Результат выражения

$$4 \cdot 625^9 - 25^{15} + 2 \cdot 5^{11} - 7$$

записали в 5-ричной системе счисления. Сколько цифр 4 в получившейся записи?

Решение 1, аналитическое:

Переведем выражение в степенной ряд 5.

$$4 \cdot 5^{36} - 5^{30} + 2 \cdot 5^{11} - 2 \cdot 5 + 3$$

Для такого выражения в пятеричной система счисления четверки будут в разрядах 35-30 и 10-2.

Следовательно, таких разрядов $(35-30+1) + (10-2+1) = 15$.

Ответ: 15

Решение 2, программное.

Python

```
x = 4*625**9 - 25**15 + 2*5**11 - 7
c = 0
while x > 0:
    c += (x % 5 == 4)
    x //= 5
print(c)
```

PascalABC.net

```
var x, c: biginteger;
begin
    x := 4*625bi**9 - 25bi**15 + 2*5bi**11 - 7;
    c := 0;
    while x > 0 do begin
        if x mod 5 = 4 then c := c + 1;
        x := x div 5;
    end;
    writeln(c);
end.
```

Увы, на C++ нет встроенной реализации больших чисел.

Ответ: 15

15 Найдите максимальное значение параметра A , при котором выражение

$$(2x + y \neq 70) \vee (x < y) \vee (A < x)$$

истинно (т.е. принимает значение 1) при любых неотрицательных значениях x и y .

Решение 1, аналитическое переборное

Значение параметра A влияет на значение всего выражения в тех случаях, когда первые два подвыражения не выполняются.

Или

$$\begin{cases} (2x + y \neq 70) = \text{Ложь} \\ (x < y) = \text{Ложь} \end{cases} \quad \text{или} \quad \begin{cases} 2x + y = 70 \\ x \geq y \end{cases}$$

Переберем все пары целых чисел (x, y) , когда это условия соблюдаются.
 (35, 0), (34, 2), (33, 4), (32, 6), (31, 8), (30, 10), (29, 12), (28, 14), (27, 16),
 (26, 18), (25, 20), (24, 22).

Заметим, что минимальное значение x в этих парах – 24.

Ответ: 23

Решение 2, аналитическое графическое

Найдем точки, соответствующие системе.

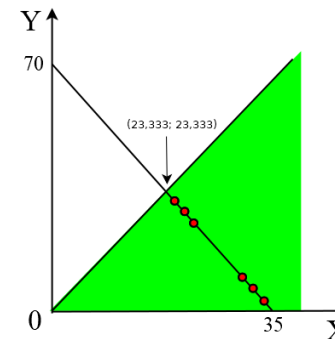
$$\begin{cases} 2x + y = 70 \\ x \geq y \end{cases}$$

$$70 - 2x = x$$

$$3x = 70$$

$$x = \frac{70}{3} = 23\frac{1}{3}$$

Найдем эти точки на плоскости



Возьмем точку с минимальным значением x – (24, 22).

Ответ: 23.

Решение 3, программное.

Python
<pre>for A in range(100, -1, -1): if all((2*x+y!=70) or(x < y) or(A < x) for x in range(100) for y in range(100)): print(A) break</pre>
PascalABC.net
<pre>var a, x, y: integer; fl: boolean; begin for a := 100 downto 0 do begin fl := true; for x := 0 to 100 do for y := 0 to 100 do if not((2*x+y <> 70) or(x < y) or(A < x)) then fl := false; if fl then begin writeln(a); break; end; end; end.</pre>
C++
<pre>#include <iostream> using namespace std; int main(){ int a, x, y; bool fl; for(a=100; a>=0; a--){ fl = true; for(x=0;x<=100; x++) for(y=0;y<=100;y++) if(!((2*x+y != 70) (x<y) (a<x))) fl = false; if(fl){ cout << a; break; } } }</pre>

Ответ: 23

16

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

$F(n) = 0$, при $n \leq 1$

$F(n) = F(n-1) + 3n^2$, при $n > 1$ и n - нечетно

$F(n) = n/2 + F(n-1) + 2$, при $n > 1$ и n - четно

Чему равно значение функции $F(49)$? В ответе запишите только целое число.

Решение, программное

Python
<pre>def F(n): if n <= 1: return 0 if n % 2 == 1: return F(n-1) + 3*n*n return n // 2 + F(n-1) + 2 print(F(49))</pre>
PascalABC.net
<pre>function F(N: integer): integer; begin if n <= 1 then F := 0 else if n mod 2 = 1 then F := F(n-1) + 3*n*n else F := (n div 2) + F(n-1) + 2 end; begin writeln(F(49)) end.</pre>
C++
<pre>#include <iostream> int F(int n){ if(n <= 1) return 0; if(n % 2 == 1) return F(n-1) + 3*n*n; return n/2 + F(n-1) + 2; } int main(){ std::cout << F(49);}</pre>

Ответ: 62820

17 Рассматривается множество целых чисел, принадлежащих числовому отрезку [12972; 89322], которые при делении на 13 дают остаток 7, при этом не делятся ни на 7, ни на 11.

Найдите наибольшее из таких чисел и их количество.

В ответе укажите два числа – сначала количество найденных чисел, затем наибольшее найденное число.

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

--	--

Решение, программное

Python
<pre>c, m = 0, 0 for i in range(12972, 89323): if i % 13 == 7 and i % 7 != 0 and i % 11 != 0: c, m = c + 1, i print(c, m)</pre>
PascalABC.net
<pre>var c, m, i: integer; begin c := 0; m := 0; for i := 12972 to 89322 do if (i mod 13=7)and(i mod 7<>0)and(i mod 11 <>0) then begin c := c + 1; m := i; end; writeln(c, ' ', m); end.</pre>
C++
<pre>#include <iostream> int main(){ int c = 0, m = 0; for(int i = 12972; i <= 89322; i = i + 1) if(x%13==7 and x%7!=0 and x%11 != 0){ c = c + 1; m = i; } std::cout << c << ' ' << m; }</pre>

Ответ: 4576 89317

19

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 четыре камня, или увеличить количество камней в куче вдвое. Например, из кучи в 15 камней игрок может получить кучу из 16, 19 или 30 камней. Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в кучах становится не менее 40. Победителем считается игрок, сделавший последний ход. В начальный момент в куче было S камней, $1 \leq S \leq 39$.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника.

Известно, что Петя не может выиграть своим первым ходом, однако после любого хода Пети Ваня может выиграть. При каком значении S это возможно?

20

Известно, что Петя имеет выигрышную стратегию в два хода, при этом Петя не может выиграть первым ходом. Укажите два значения S , при которых это возможно. Значения укажите в порядке возрастания.

21

Известно, что Ваня имеет выигрышную стратегию за один или два хода, при этом не имеет выигрышной стратегии в один ход. Найдите минимальное значение S , при котором это возможно.

Решение, аналитическое

Сперва найдем, из каких позиций можно выиграть первым ходом (сразу). Это значения из диапазона $[20;39]$. Увеличив кучу вдвое, игрок получает количество камней, достаточное для победы.

19.

Поэтому если игрок любым своим ходом приходит в одну из позиций выше, то он проигрывает. Такое значение S всего одно – 19.

Ответ: 19

20.

Чтобы игрок победил вторым своим ходом, необходимо свести игру к значению 19. Это можно сделать из позиций 18 и 15.

Ответ: 15 18

21.

Чтобы противник победил первым или вторым ходом при правильной игре, необходимо чтобы все ходы текущего игрока приходили либо в одно из значений из п.20, либо в $[20;39]$.

Таких значения два – 14 и 17.

Ответ: 14

22

Ниже записана программа. Получив на вход число x , эта программа печатает два числа a и b . При каком наименьшем значении x после выполнения программы на экран будет выведено два числа 10, а затем 6.

C++	Паскаль
<pre>#include <iostream> using namespace std; int main(){ int a = 0, b = 0, x; cin >> x; while (x > 0) { int c = x % 10; a = a + c; if(b < c) b = c; x = x / 10; } cout << a << endl << b; return 0; }</pre>	<pre>var x, c, a, b: longint; begin readln(x); a := 0; b := 0; while x > 0 do begin c := x mod 10; a := a + c; if b < c then b := c; x := x div 10; end; writeln(a); write(b); end.</pre>
Python	Алгоритмический язык
<pre>x = int(input()) a, b = 0, 0 while x > 0: c = x % 10 a = a + c if b < c: b = c x = x // 10 print(a) print(b)</pre>	<pre>алг нач цел x, a, b, c a := 0 b := 0 нц пока x > 0 c := mod(c, 10) a := a + c если b < c то b := c конец если x := div(x, 10) кц вывод a вывод b кон</pre>

Ответ: _____.

Решение, аналитическое

Алгоритм перебирает в переменной c значения разрядов числа. Причем в a сохраняется сумма всех разрядов, в b – максимальное значение.

Следовательно, мы имеем число c суммой разрядов 10 и максимальным разрядом 6. Минимальное число с такими характеристиками – 46.

Ответ: 46

23 Исполнитель преобразует число, записанное на экране. У исполнителя есть две команды, которым присвоены номера:

1. Прибавить 1
2. Прибавить 2
3. Умножить на 3

Первая команда увеличивает число на 1, вторая – на 2, третья - втрое.

Программа для исполнителя – это последовательность команд. Сколько существует таких программ, которые исходное число 2 преобразуют в число 19 и при этом траектория вычислений программы проходит через 9 и не проходит через 12?

Решение аналитическое

Построим две таблицы – от 2 до 9, и от 9 до 19.

2	3	4	5	6	7	8	9
1							

9	10	11	12	13	14	15	16	17	18	19
1			0							

И найдем количество траекторий из 2 в 9 и из 9 в 19.

2	3	4	5	6	7	8	9
1	1	2	3	6	9	15	25

9	10	11	12	13	14	15	16	17	18	19
1	1	2	0	2	2	4	6	10	16	26

Для получения ответа перемножим значения для 9 и 19.

Ответ: 650

Решение программное

Python
<pre>def f(a, b): if a == b: return 1 if a > b or a == 12: return 0 return f(a+1, b) + f(a+2, b) + f(a*3, b) print(f(2,9) * f(9, 19))</pre>
PascalABC.net
<pre>function f(a, b: integer): integer; begin if a = b then f := 1 else if (a > b) or (a = 12) then f := 0 else f := f(a+1, b) + f(a+2, b) + f(a*3, b); end; begin writeln(f(2, 9)*f(9, 19)) end.</pre>
C++
<pre>#include <iostream> int f(int a, int b){ if(a == b) return 1; if(a > b or a == 12) return 0; return f(a+1, b) + f(a+2, b) + f(a*3, b); } int main(){ std::cout << (f(2, 9)*f(9, 19)); }</pre>

Ответ: 650

Задание выполняется с использованием прилагаемых файлов.

24

Текстовый файл состоит не более, чем из 10^7 строчных букв английского алфавита.

Найдите максимальную длину подстроки, в которой символы a и d не стоят рядом.

Решение программное с разбиением на слова

Описание: заменяем все вхождения 'da' и 'ad' на строки 'd a' и 'a d'. Таким образом между любыми рядом стоящими a и d будет пробел. После этого разделяем строку по пробелам и находим максимальную длину получившихся подстрок.

Python

```
f = open('24.txt')
s = f.readline() \
    .replace('ad', 'a d') \
    .replace('da', 'd a')
w = list(map(len, s.split()))
print(max(w))
```

PascalABC.net

```
var f: text;
    s: string;
    ar_s: array of string;
    m, i: integer;
begin
    assign(f, '24.txt');
    f.reset();
    readln(f, s);
    s := s.replace('da', 'd a').replace('ad', 'a d');
    m := 0;
    ar_s := s.split();
    for i := 0 to ar_s.Length - 1 do
        if ar_s[i].Length > m then
            m := ar_s[i].Length;
    writeln(m);
end.
```

На C++ можешь сделать по аналогии, по мне проще сразу делать алгоритм ниже =)

Ответ: 2252

Решение программное за одно чтение

Описание: минимальная длина последовательности равна 1, потому что строка из одного символа не может содержать рядом стоящих d и a. Начинаем проверять от второго символа в строке. Если предыдущий и текущий символы в строке не образуют строку 'da' или 'ad', увеличиваем длину текущей подстроки на 1, иначе начинаем новую подстроку. После каждого изменения длины проверяем на максимум.

Python

```
f = open('24.txt')
s = f.readline()
c, m = 1, 0
for i in range(1, len(s)):
    if s[i-1:i+1] not in ['da', 'ad']:
        c += 1
    else:
        c = 1
    m = max(m, c)
print(m)
```

PascalABC.net

```
var f: text;
    s: string;
    c, m, i: integer;
begin
    assign(f, '24.txt');
    f.reset(); readln(f, s);
    c := 1; m := 0;
    for i := 2 to s.Length do begin
        if not(s[i-1:i+1] in ['da', 'ad']) then
            c := c + 1
        else c := 1;
        m := max(m, c);
    end;
    writeln(m);
end.
```

C++

```
#include <iostream>
#include <fstream>
using namespace std;
int main(){
    ifstream f("24.txt");
    string s;
    f >> s;
    int c = 1, m = 0;
    for(int i=1; i<s.length(); i++){
        if(!(s[i-1] == 'd' && s[i] == 'a' ||
            s[i-1] == 'a' && s[i] == 'd'))
            c = c + 1;
        else
            c = 1;
        if(m < c) m = c;
    }
    std::cout << m;
}
```

Ответ: 2252

25

Найдите 5 чисел больших 500000, таких, что среди их делителей есть число, оканчивающееся на 8, при этом этот делитель не равен 8 и самому числу.

В качестве ответа приведите 5 наименьших чисел, соответствующих условию.

Формат вывода: для каждого из 5 таких найденных чисел в отдельной строке сначала выводится само число, затем минимальный делитель, оканчивающийся на 8, не равный 8 и самому числу.

Решение

Python

```

c = 0
for i in range(500000, 10000000):
    md = 0
    for d in range(2, int(i**0.5) + 1):
        if i % d == 0:
            if d % 10 == 8 and d != 8:
                md = d
                break
            if (i // d) % 10 == 8:
                md = i // d
    if md > 0:
        print(i, md)
        c += 1
        if c == 5:
            break

```

PascalABC.net

```

var c, d, i, md: integer;
begin
    for i := 500000 to 10000000 do begin
        md := 0;
        for d := 2 to trunc(sqrt(i)) do
            if i mod d = 0 then begin
                if (d mod 10 = 8) and (d <> 8) then begin
                    md := d;
                    break;
                end;
                if (i div d) mod 10 = 8 then
                    md := i div d;
            end;
        if md > 0 then begin
            writeln(i, ' ', md);
            c := c + 1;
            if c = 5 then break;
        end;
    end;
end.

```

C++

```

#include <iostream>
#include <cmath>
using namespace std;
int main(){
    int c = 0;
    for(long int i=500000; i<10000000; i++){
        int md = 0;
        for(int d = 2; d <= round(sqrt(i)); d++){
            if(i % d == 0){
                if(d != 8 && d % 10 == 8){
                    md = d;
                    break;
                }
                if((i / d) % 10 == 8)
                    md = i / d;
            }
        if(md > 0){
            std::cout << i << " " << md << std::endl;
            c = c + 1;
            if(c == 5) break;
        }
    }
}

```

Ответ:

500002 178

500004 18

500016 48

500018 58

500020 4348



Задание выполняется с использованием прилагаемых файлов.

26

Организация купила для своих сотрудников все места в нескольких подряд идущих рядах на концертной площадке. Известно, какие места уже распределены между сотрудниками. Найдите ряд с наибольшим номером, в котором есть два соседних места, таких что слева и справа от них в том же ряду места уже распределены (заняты). Гарантируется, что есть хотя бы один ряд, удовлетворяющий условию. В ответе запишите два целых числа: номер ряда и наименьший номер места из найденных в этом ряду подходящих пар.

Входные данные.

В первой строке входного файла находится одно число:

N – количество занятых мест (натуральное число, не превышающее 10 000).

В следующих N строках находятся пары чисел: ряд и место выкупленного билета (числа не превышают 100 000).

Выходные данные.

Два целых неотрицательных числа:

Максимальный номер ряда, где нашлись обозначенные в задаче места и минимальный номер места.

Пример входного файла:

6
50 12
50 15
60 157
60 160
60 22
60 25

Выходные данные:

60	23
----	----

Ответ:

--	--

Решение:

- 1) Считаем все пары в двумерный массив, где первое число – номер ряда, второе – номер места.
 - a. Номер места считаем как отрицательное значение, чтобы при сортировке места с меньшим номером (по данным из файла) были больше (в данных). Это существенно сократит поиск нужного места.
- 2) Отсортируем массив.
- 3) Найдем последний элемент в отсортированном массиве, который удовлетворяет условию.
 - a. Два занятых места находятся в одном ряду,
 - b. Разница между проверяемым местом и предыдущим равна 3 (соответствует схеме «занято»-«свободно»-«свободно»-«занято»).
 - c. Так как надо определить минимальный номер свободного места, добавим к номеру найденного места 1 (соседнее справа от минимального значения в паре занятых).

Python

```
f = open('26.txt')
n = int(f.readline())
nums = []
for _ in range(n):
    pair = list(map(int, f.readline().split()))
    pair[1] = -pair[1]
    nums += [pair]
nums.sort()
r, m = 0, 0
for i in range(1, len(nums)):
    if nums[i][0] == nums[i-1][0]:
        if nums[i][1] - nums[i-1][1] == 3:
            r = nums[i][0]
            m = -nums[i][1] + 1
print(r, m)
```

PascalABC.net

```
var f: text;
    n, i, r, m, a, b: integer;
    nums: array of array of integer;
begin
    setlength(nums, 0);
    assign(f, '26.txt');
    reset(f);
    readln(f, n);
    loop n do begin
        readln(f, a, b);
        nums := nums + ||a, -b||;
    end;
    Sort(nums, (x, y) -> ((x[0] < y[0]) or (x[0] =
y[0]) and (x[1] < y[1])));
    r := 0; m := 0;
    for i := 1 to nums.Length - 1 do
        if nums[i, 0] = nums[i-1, 0] then
            if nums[i, 1] - nums[i-1, 1] = 3 then begin
                r := nums[i, 0];
                m := -nums[i, 1] + 1;
            end;
    end;

    print(r, m);
end.
```

C++

```
#include <iostream>
#include <algorithm>
#include <fstream>
#include <vector>
using namespace std;
int main(){
    ifstream f("26.txt");
    int n, a, b, r, m;
    f >> n;
    vector <vector <int>> nums;
    for(int i=1; i<=n; i++){
        f >> a >> b;
        vector<int> temp = {a, -b};
        nums.push_back(temp);
    }
    sort(nums.begin(), nums.end());
    r = 0; m = 0;
    for(int i=1; i < nums.size(); i++)
        if(nums[i][0] == nums[i-1][0])
            if(nums[i][1] - nums[i-1][1] == 3){
                r = nums[i][0];
                m = -nums[i][1] + 1;
            }
    cout << r << " " << m;
}
```

Ответ: 8631 7311



Задание выполняется с использованием прилагаемых файлов.

27

На вход программы поступает последовательность из целых положительных чисел. Необходимо выбрать такую подпоследовательность подряд идущих чисел, чтобы их сумма была максимальной и делилась на 89, а также её длину. Если таких подпоследовательностей несколько, выбрать такую, у которой длина меньше.

Входные данные.

Даны два входных файла (файл А и файл В), каждый из которых содержит в первой строке количество чисел N ($2 \leq N \leq 10^8$). В каждой из последующих N строк записано одно целое положительное число, не превышающее 10000. Программа должна вывести длину найденной последовательности.

Пример организации исходных данных во входном файле:

```
8
2
3
4
93
42
34
5
95
```

Для делителя 50 при указанных входных данных значением искомой суммы должно быть число 100 (3+4+93 или 5+95). Следовательно, ответ на задачу – 2.

В ответе укажите два числа: сначала длину искомой подпоследовательности для файла А, затем для файла В.

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

Решения:

Переборное

Перебираем все возможные суммы, которые начинаются на i -том элементе и заканчиваются на j -том элементе. Нашли БОльшую сумму – обновили ms (макс.сумму) и m (мин.длину). Нашли такую же сумму, как уже найденная максимальная – обновили m при необходимости.

```
Python
f = open('27_A.txt')
n = int(f.readline())
nums = list(map(int, f.readlines()[1:n]))
n, ms = float('inf'), 0
for i in range(len(nums)):
    for j in range(i, len(nums)):
        s = sum(nums[i:j + 1])
        if s % 89 == 0 and s > ms:
            ms, m = s, j - i + 1
        if s % 89 == 0 and s == ms:
            m = min(m, j - i + 1)
print(m)
```

*Динамическое***Python**

```
f = open('27_B.txt')
k, s = 89, 0
mins = {0: (0, 0)}
res = []
for i in range(1, int(f.readline())+1):
    s += int(f.readline())
    if s % k in mins:
        res += [(s - mins[s % k][0],
                mins[s % k][1] - i)]
    else:
        mins[s % k] = (s, i)
print(-max(res)[1])
```

*Через метод частичных сумм***Python**

```
f = open('27_B.txt')
n = int(f.readline())
k = 89
r = {0: (0, 0)}
ms = 0
m = float('inf')
for _ in range(n):
    x = int(f.readline())
    t = {}
    for key in r:
        t[(key+x) % k] = (r[key][0] + x, r[key][1] +
1)
    if x % k not in t:
        t[x % k] = (x, 1)
    r = t.copy()
    if 0 in r:
        if ms < r[0][0]:
            ms = r[0][0]
            m = r[0][1]
        elif ms == r[0][0]:
            m = min(t[0][1], m)
print(m)
```

Ответ:

159	235660
-----	--------

Система оценивания экзаменационной работы по информатике и ИКТ

За правильный ответ на задания 1–24 ставится 1 балл; за неверный ответ или его отсутствие – 0 баллов.

За верный ответ на задание 25 ставится 2 балла; за ошибочные значения только в одной строке ответа ИЛИ за отсутствие не более одной строки ответа ИЛИ присутствие не более одной лишней строки ответа ставится 1 балл. В остальных случаях – 0 баллов.

За верный ответ на задание 26 ставится 2 балла; если значения в ответе перепутаны местами ИЛИ в ответе присутствует только одно верное

значение (второе неверно или отсутствует) – ставится 1 балл. В остальных случаях – 0 баллов.

За верный ответ на задание 27 ставится 2 балла; если значения в ответе перепутаны местами ИЛИ в ответе присутствует только одно верное значение (второе неверно или отсутствует) – ставится 1 балл. В остальных случаях – 0 баллов.

Файлы к варианту: <https://vk.cc/c3geii>

Ссылка на тест в эмуляторе: <https://vk.cc/c3ktoG>

Информация об авторе

Автор	Евгений Джобс vk.com/eugenyjjobs
Группа проекта	vk.com/inform_web
Канал на youtube	www.youtube.com/c/EvgenijJobs
Автор эмулятора	vk.com/cabanovalexey
Канал на youtube	www.youtube.com/user/axelofan2010

№ задания	Ответ	
1	21	
2	cdba	
3	4	
4	1011	
5	56	
6	24	
7	64	
8	2200	
9	49	
10	5	
11	330	
12	881	
13	16	
14	15	
15	23	
16	62820	
17	4576	89317
18	761	579
19	19	
20	15	18
21	14	
22	46	
23	650	
24	2252	
25	50002	178
	50004	18
	50016	48
	50018	58
	50020	4348
26	8631	7311
27	159	235660