

Олимпиада школьников «Надежда энергетики»

№ группы

№ группы

Вариант № 7112

01 61-73

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ АБРАМОВ

ИМЯ ДМИТРИЙ

ОТЧЕСТВО ПАВЛОВИЧ

Дата рождения 21.01.1998

Класс: 11

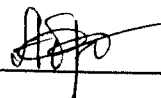
Предмет Информатика

Этап: Заключительный

Работа выполнена на 5 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№1

Виды др. $\frac{1}{10}$

1) представить данное число в виде суммы его разрядов. Например: $521 = \underbrace{500}_{\text{сотни}} + \underbrace{20}_{\text{десят.}} + \underbrace{1}_{\text{единиц.}}$

$9383 = \underbrace{9000}_{\text{тысяч}} + \underbrace{300}_{\text{сотни}} + \underbrace{80}_{\text{десят.}} + \underbrace{3}_{\text{единиц.}}$

2) Далее переводим в римскую сс. каждое из слагаемых, начиная от большего разряда к меньшему. Тригем: переводим первый (старший разряд), выписываем ответ, далее переводим следующий и результатом приписываем к тому, что мы получили до этого. так делаем, пока всё не проверим.

Например: $521 = 500 + 20 + 1$

$500 \rightarrow \text{D} \rightarrow \text{D}$ (тригем)
 $20 \rightarrow \text{XX} \rightarrow \text{DXX}$
 $1 \rightarrow \text{I} \rightarrow \text{DXXI}$



3) Теперь обсудим сам процесс перевода:

Мы знаем, что $\text{I} = 1$ $\text{L} = 50$
 $\text{V} = 5$ $\text{C} = 100$ $\text{M} = 1000$
 $\text{X} = 10$ $\text{D} = 500$

Если число (разряд) меньше равен N .
 Нужно найти такую букву Р сс, которая будет равна или больше N (ближайшая к нему).
 Тогда выписываем эту букву, а из N вычитаем значение этой буквы в 10-ой сс. повторим это, пока $N < 0$.

Пример: Если $N =$ (буква большего числа - буква меньшего числа), но и записать нужно так же (меньшая буква перед большей).
 Например: $400 = 100 + 100 + 100 + 100 - \text{ай}$.
 $400 = (500 - 100) = \text{CD}$.



т.е. если в записи числа (римской с.с.) буква, у которой значение меньше, стоит перед буквой, у которой значение больше (IX), то надо из знач. большей вычесть меньшее, а если наоборот (XI), то сложить.

№2

а) Запись рационального числа в (2-10) с.с. целесообразна тогда, когда нам не важен объект информации, которая эта запись записывает. Верь в десятичной системе счисления, запись (десяти) числа в 2-ой с.с. меньше, чем запись (десяти) в (2-10)-ой.

$$i_1 = \log_2 a \quad i_1 < i_2, \text{ где } a - \text{кол-во цифр в записи числа в 2-с.с.}$$

$$i_2 = \log_2 (b \cdot 4) \quad \uparrow \quad \downarrow$$

↑ кол-во цифр в записи 2-10 ↓ кол-во цифр в записи 10-ой с.с.

Например; $5_{10} = 10011_2 = 0011_3 0001_4 0001_{2-10}$.

$$a = 9 \quad \log_2 9 < \log_2 12$$

$$b = 4 \cdot 3 = 12$$

Ответ: • В случае, когда не важно кол-во цифр.
• В 2-10 с.с. легче выполнить операции над числами (+, -, °, :)

№3.

1) Вводим (n) и (a).

2) Проверяем, является ли он простым числом (если нет, то выводим "Не существует", конец программы).

Если является, всё дальше.



3) перебираем b от 0 до $(n-1)$. $(a \cdot b \bmod n = 1)$
 проверяем выражение $a \cdot b \equiv 1 \pmod{n}$. $\frac{a \cdot b}{n} = \dots, 1$
 если такое b существует, то выводим его,
 если $b = n-1$, а результата нет, то
 выводим "не существует".
 конец.

ИЧ

1) Вводим N -кол-во квадратов.
 2) если $N=1$, вывести 4
 если $N=2$, вывести 7
 3) если $N \geq 3$.
 $N := N - 2$ (т.к. 2 квадрата взяли в ~~смысле~~ ^{смысле} n_2)
 $K := 7$ (кол-во рек). (смысл n_2)
 $a := 1$ (сколько раз нужно прибавить 2 рейки перед поворотом). (сколько квадратов перед поворотом).
 Итак, на черновике я вывел такую закономерность:
 кол-во: $(3 + 2 \cdot a + 3 + 2 \cdot a)$ - кол-во рек, после
 которых a увеличивается на 1. $(a+1)$.
 т.е. кол-во квадратов $= (1 + a + 1 + a) = (2a + 2)$

4) Если $N - (2a + 2) \geq 0$, то
 ~~$N := 2a + 2$~~ $N := N - (2a + 2)$;
 $K := 3 + 3 + 2a + 2a$;
 $a := a + 1$.

Повторим условие (4) пока это возможно.

~~5) если $N=0$, то конец программы.~~

~~если $N < 0$, то конец!~~

~~если $K < 0$, то~~

~~$N := N - 1$;~~

~~$K := K + 3$;~~

~~если $N < 0$~~

~~$N := N - 1$~~

(продолжение на
 другом листе)



Выполнил доп. работ
17
Томас

5) если $N = 0$, то конец программы.

6) если $N > 0$, то

$$N := N - 1;$$

$$K := K + 3;$$

• повторим а раз пока $N > 0$

$$N := N - 1;$$

$$K := K + 2;$$

• если $N > 0$
 $N :=$



7) повторим яичко пункт (6) пока $N < 0$.

8) Возвести K .

НБ

1) Как хранить?

заведем 3 массива (2 массива и 1 массив)

а - хранения и П/Л

б - значение парашюта

с - результатом.

2) Как работать?

задача: упорядочить по возрастанию значений результата.

Значит, будем упорядочивать массив

с, а остальные массивы (а, б) просто дублируем.

3) как упорядочить массив с по возрастанию значения на месте НБ.



3) Упорядочить массив с (результат) *Вопрос-реш. лист № 2*
по возрастанию. *Уксус*

Берём первый элемент и сравниваем с остальными. Если какой-то элемент меньше его, то меняем их местами. Берём 2-ой элемент и сравниваем со всеми остальными, потом 3-ий и так далее до конца массива. F

Но забываем в это время «сдвигать» массивы a и b .

В результате выведет все массивы.

№6

Считаем сумму всех парней y каждого уровня.

y те́тра - 2221 (за все ^{этаж} парней)
 y Даша - 2222

если $2221 > 2222$ выведи «тётра»

если $2222 > 2221$ выведи «Даша»

если $2221 = 2222$ выведи «ничья»

F

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7092

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Андреева

ИМЯ Влада

ОТЧЕСТВО Алексевна

Дата рождения 04.08.1999

Класс: 9

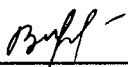
Предмет информатика

Этап: защитительный

Работа выполнена на 3 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№3.

Kog Pascal.

```

var n, k, f, i, j, min, ost, nod: longint;
var w, p, l, j, nod: longint; begin
readln(p); w
w := 0;
for i := 1 to (p-1) do
begin
for j := 1 to i do
if ((p mod j = 0) and (i mod j = 0)) then nod := j;
if (nod = 1) then w := w + 1;
end;
writeln(w);
end.

```



№4. Kog Pascal.

var
begin~~read~~

var a, b: array[1..1000000000] of longint;

n, m, n1, j, i, d, d1, k, c: longint;

begin

readln(n, m);

n1 := n; j := 0; d := 0; k := 0;

~~for i := 1 to~~ while (n1 <> 0) do

begin

n1 := n1 div 10;

k := k + 1;

end;

n1 := n;

for i := 1 to k do

begin

a[i] := n1 mod 10;

b[i] := a[i];

n1 := n1 div 10;

d := d + 1;

end;

~~c := a[1];~~ c := 0;

for i := 1 to m do

begin

d1 := d;

d := 0;

while (d1 <> 0) do





```
begin
  c := c + (a[j] * b[i]);
  a[j] := c mod 10;
  c := c div 10;
  d1 := c;
  d := d + 1;
end;
end;
for i := d downto 1 do write (a[i]);
end.
```

№5. Код C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
  for (long int i = 10000; i < 1000000000; ++i) {
    if ((i & 7 == 0) && (i & 11 == 0)) cout << i << " ";
```

```
  }
  return 0;
```

```
}
```

№7. Код Pascal.

```
var a: array[1..10000] of longint;
```

```
    d, c, i, j: longint;
```

```
begin
```

```
  c := 0;
```

```
  a[1] := 1;
```

```
  for i := 2 to 44 do
```

```
    begin
```

```
      j := 0; d := 1; c := 0;
```

```
      while (d <> 0) do
```

```
        begin
```

```
          j := j + 1;
```

```
          c := c + (a[j] * i);
```

```
          a[j] := c mod 10;
```

```
          c := c div 10; d := c;
```

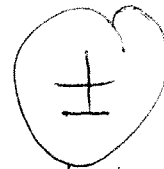
```
        end;
```

```
      end;
```

```
    for i := j downto 1 do
```

```
      if (a[i] = 4) then write (i, ' ');
```

```
end.
```





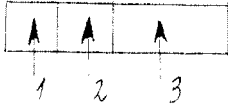
№1.

$$\overset{4}{a}\overset{3}{b}\overset{2}{c}\overset{1}{d}_f = a \cdot 4! + b \cdot 3! + c \cdot 2! + d \cdot 1!$$

Факториалы или назовем множителями. (+)

Эта система счисления является позиционной, т.к. каждый множитель зависит от разряда, что соответствует определению позиционной системы счисления.

№2.



1. Переводят ли в другую сторону земли? (-)
2. Сладкие один из дней с зимним временем?
3. Сладкие один из дней с летним временем?

Если $\begin{cases} 101, \\ 010, \\ 001, \end{cases}$ то GMT-дачное
 $\begin{cases} 110, \end{cases}$ то GMT-дачное + 1 час.

№6.

При работе на компьютере складываются двоичные представления чисел.



Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7112

ФАМИЛИЯ АХУНОВА

ИМЯ РЕГИНА

ОТЧЕСТВО ИЛЬДАРОВНА

Дата рождения 22.08.1997

Класс: 11


Предмет Информатика

Этап: Заключительный

Работа выполнена на 2 листах

Дата выполнения работы: 24.02.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



1. Нам число \rightarrow число в римской с.с. (I, V, X, L, M...)

Алгоритм, для чисел < 4 число записывается с помощью
 букв «I», где ко-во = количеству
 Если число = 4 \Rightarrow запись состоит из «I», затем «V» (т.е. ко-во «I» = 4),
 сколько не хватает числу до 5.
 Далее для чисел от 5 до 9, а затем столько «I», сколько со-ед.
 Аналогично для «X» для чисел ≥ 9 запись «I» в таком кол-ве,
 а затем «X». Для чисел > 5 вы-ся в от-ном по-ке, ~~и т.д.~~
 Для разрядов десятков вы-ся след-ее: число разн. цифра в
 разряде, столько в запись числа букв «X» и так далее.
 $I = 1, V = 5, X = 10, L = 50, M = 100 \dots$

2. Число 311 в 2-с.с. $311_{10} = 1.0011.0111_2$

(т.к. 311_2
 $\begin{array}{r} 155 \\ \cdot 2 \\ \hline 1 \\ \cdot 1 \\ \hline 77 \\ \cdot 2 \\ \hline 1 \\ \cdot 1 \\ \hline 38 \\ \cdot 2 \\ \hline 1 \\ \cdot 1 \\ \hline 19 \\ \cdot 2 \\ \hline 1 \\ \cdot 1 \\ \hline 9 \\ \cdot 2 \\ \hline 4 \\ \cdot 2 \\ \hline 2 \\ \cdot 2 \\ \hline 1 \end{array}$)

и в 10-с.с.

$311_{10} = 0011.0001.0001_{10}$ (т.к. в 4-с.с. $3 = 0011, 1 = 0001$)

Применение двоично-десятичной сис. сис. целесообразно, если:

- 1) На каждую позицию в алфавите (в цифрах) выделяется ровно 4 знака (□□□□).
- 2) Если какой-либо символ алфавита переводится в двоичную сис. сис. переводится в двоичную сис. сис. переводится в двоичную сис. сис.
- 3) Преобразование можно производить переводом из 2-ой с.с. в 4-ую с.с. сис. сис. переводится в двоичную сис. сис.
- 4) Эти вычисления алфавит действий с числами.

3. Вводим число a ($0 \leq a < n$) и значение модуля n . Проверка, является ли число a и n взаимно простыми. Для этого разлагаем a и n на простые множители (2, 3, 5, 7, ...). Если это условие выполнено и число взаимно просто, тогда можно взаимно однозначно обратное число $b = n - a$. Т.к. должно вы-ся след-ее:
 $(n - b) \cdot a = 0 \Rightarrow n - b = a \Leftrightarrow b = n - a$
 иначе взаимно однозначно обратного числа не существует для данного числа a по модулю n .



4. Пусть N -ка-во квадратов. (N :integer, $N > 0$)

Создай массив $a[1..N]$;

S -ка-во пеек;

Изначально, $S=4$. if $(i=1)$ and $(N=1)$ then $S=4$. $W=N(S)$
(т.к. где 1 квадрат 4 пеек)

for $i=2$ to N do begin

if $((i=2)$ or $(i \bmod 2=1))$ then $S:=S+3$;

if $((i \bmod 2=0)$ and $(i > 2))$ then $S:=S+2$;

end;

Так; программа выведет S -массив. ка-во пеек, необход-во
проверка: $N=5$; $i=1$: $S=4$; $i=2$: $S=7$; $i=3$: $3 \bmod 2=1$ $S=10$; $i=4$: $4 \bmod 2=0$ $S=12$
 $i=5$: $5 \bmod 2=1 \Rightarrow S=15$ (что упр-еи дан-ый результат)

5. Необходимо создать базу данных мб в программе M.Evel,
мб в про-ме M.Acess.

Записать форму так, как сказано по усл-ю
Земли, выделенной шлобам "результат", и задать в парам-тр
базы данных, выбрать условие по выделению
(от 1-го к последнему). При этом, закрепив всю сумму
всех земель, чтобы переключалась на одну ячейку в таблице,
а все строки (на, парам-тр результат).

Зем создат двумерный массив в программе Pascal $a[i,j]$
 i -строка матрицы $a[i,1]$ - по п/н, $a[i,2]$ - значения элементов
из ряда функции $assign$, $reset$ и массивы str (массив
 $a[i,3]$ - строки результатов (записывается аналогично)
в строке i и j)
Затем инициализировать строки по массиву for $i=2$ to N
(N -номер кол-ва строк): if $a[i,3] < a[i,3]$ then begin
 $a[i,3] := a[i,3]$; $a[i,2] := a[i,2]$; end;
Таким образом, массивы массив. результатов. Аналогично,
произведем сравнение с a -ой $2..N$.

6. Т.к. по условию матрица заданной ширины не дана,
след-но предположим ее сами:

будем считать, что на странице 3 столбце
(N_3 , $рез-т1$, $рез-т2$) а ка-во строк = M .
Для проверки шлоб, необходимо знать, как резуль-
таты даны-и 1-ой, а кей 2-ой (хоть $рез-т1$,
 $рез-т2$). Тогда, окончательно результат $рез-т1$

$P = \sum_{k=1}^M X(M,2)$, т.е. сумма всех клеток строки 2 .

а результат данна: $D = \sum_{k=1}^M Y(M,3)$; (сумма всех клеток 3 -ой)

В итоге получим тот, кто $какой$ больше очков. Если $P > D$
то write ('Temp'), а если $D > P \Rightarrow$ write ('Данна');
если $P=D$ write ('Ничья')

Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7112

ФАМИЛИЯ

Баранов

ИМЯ

Юрий

ОТЧЕСТВО

Александрович

Дата
рождения

22.12.97

Класс:

11

Предмет

информатика

Этап:

заключительный

Работа выполнена на 7 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



N6.

Создадим 2 числа по краеням суммарного веса
Петя и Даша соответственно:

Var

$p, q: \text{longint}; 0..1000, 1..3$

$a: \text{array}[1..3, 0..1000] \text{ of integer};$ - блокнот строков

$i: \text{integer};$

начало

Пудим ийти вниз по блоку и прибавлять очки:

n - высота страницы, m - ширина стр.

for $i := 1$ to n do

begin

~~$p := p + a[i, 2];$~~

~~$q := q + a[i, 3];$~~

$p := p + a[i, 2];$

$q := q + a[i, 3];$

end;

if $p > q$ then

вывести ('Петя');

if $p < q$ then

вывести ('Даша');

if $p = q$ then

вывести ('Ничья');

конец.

N7. ~~$i, n: \text{integer};$~~

~~begin~~

~~вывести (n);~~

~~for $i := 1$ to $(n \text{ div } 10)$ do~~

~~write ('x');~~





N5.

Быстрее всего было бы создать 3 массива для хранения экспериментов (num, param, res) и один вспомогательный, в который записать числа от 1 до n (~~номер~~ ^{НОМЕРА} экспериментов) (a), а дальше отсортировать массив a[] по возрастанию, сравнивая ~~не a[i] и a[j]~~ res[a[i]] и res[a[j]], любой сортировкой за $O(n \log n)$, к примеру, пирамидальной.

Но так как количество экспериментов неизвестно, придется использовать списки и указатели, а так же обычно сортировку обменом.

Для удобства сделаем процедуру обмена swap.

```

Type
  point = ^list - указатель на наш список
  list = список
  num: integer;
  param: string;
  res: integer;
  end, var
} поля для данных
  n: point; ← указатель на следующий элемент списка.
procedure swap (p, q: point);
begin
  var t: integer; s: string;
начало
  t := p^.num; p^.num := q^.num; q^.num := t;
  s := p^.param; p^.param := q^.param; q^.param := s;
  t := p^.res; p^.res := q^.res; q^.res := t;
конеч.
var p, q: point;
  flag: boolean;
начало
  v := nil; l := v; // v - начало списка
  если файл не пуст, то
начало
    создать новый элемент(q);
    ввод (q^.num, q^.param, q^.res); // ввод из файла
    v := q;
    l := v;
конеч.
↓ продолжение на 4 листе

```




N1.

```
S : string;           // результат
n : integer,         // число
i : integer;
начало
  вывести(n);
  for i = 1 to (n div 10) do
    S := S + 'X';
  if n mod 10 < 4 then
    for i = 1 to n mod 10 do
      S := S + 'I';
  if n mod 10 = 4 then
    S := S + 'IV';
else
  if n mod 10 = 9 then
    S := S + 'IX';
  if (n mod 10 > 5) and (n mod 10 < 9) then
начало
    S := S + 'V';
    for i = 1 to (n mod 10 - 5) do
      S := S + 'I';
конец;
вывести(S);
конец.
```



Пока файл не пуст делать
нц

создать (q);

ввести (q¹.num, q¹.param, q¹.res);

l¹.n := q;

l := q;

кц;

flag := true;

~~while flag do~~

пока flag делать

нц

flag := false;

l := v;

пока l¹.n <> nil делать

нц

if l¹.res > l¹.n¹.res then

нц

swap(l, l¹.n);

flag := true;

кц

l := l¹.n;

кц;

l := v; ~~while~~ Пока l <> nil делать ~~нц~~ вывести (l¹.param), l := l¹.n; кц;

кц;

№2.

4 бита являясь не кодирование идет до 15 включительно, поэтому если записывать каждую цифру 10-чной системы счисления 4 битами, это приведет к существенным затратам памяти.

Можно предложить, что удобно использовать данный подход если требуется максимально быстро узнать любую цифру числа. Тогда для этого не нужно будет каждый раз переводить число, после вычисления, к примеру.

ВВОД ДАННЫХ ИЗ ФАЙЛА

сортировка

// если был совершен обмен, ставим флажок, чтобы сортировка продолжалась

ВЫВОД:



N3.

Если $ab \bmod n = 1$, то $ab - 1 = nk$, k - целое число
 Решим задачу перебором, ограничиваясь значениями k , при
 которых $b < n$

var

a, b, k, n: integer;

flag: boolean;

|| flag - метка не найдено ли значение b

~~k := 0~~~~while $(\frac{nk+1}{a} < b)$ и flag~~

нц

k := 0; ввести(n, a); flag := true;

Пока $(\frac{nk+1}{a} < b)$ и (flag) делать

нц

Если $(nk+1) \bmod a = 0$ ТО

нц

flag := false;

b := $(nk+1) \text{ div } a$;

кц;

k := k + 1;

кц;

Если flag ТО

вывести('не существует')

иначе

вывести(b);

кц.





N1.

Алгоритм заключается в том, чтобы последовательно переводить в римскую систему разряды тысяч, сотен, десятков и единицы заданного числа.

```
var n, t, i: integer
```

```
нз всех (n);
```

```
for i := 1 to n div 1000 do
```

```
  write('M');
```

```
  n := n mod 1000;
```

```
  t := n div 100;
```

```
  if t = 9 then
```

```
    write('CM');
```

```
  if (t ≥ 5) и (t < 9) then
```

```
нз
```

```
  write('D');
```

```
  for i := 1 to (t mod 5) do
```

```
    write('C');
```

```
нз
```

```
  if t = 4 then
```

```
    write('CD');
```

```
  if t < 4 then
```

```
    for i := 1 to t do
```

```
      write('C');
```

```
  n := n mod 100;
```

```
  t := n div 10;
```

```
  if t = 9 then write('XC');
```

```
  if (t < 9) и (t ≥ 5) then
```

```
нз
```

```
  write('L');
```

```
  for i := 1 to (t mod 5) do write('X');
```

```
нз
```

```
  if t = 4 then write('XL');
```

```
  if t < 4 then
```

```
    for i := 1 to t do write('X');
```

```
  t := n mod 10;
```

```
  if t = 9 then write('IX');
```

```
  if (t ≥ 5) и (t < 9) then
```

```
нз
```

```
  write('V');
```

```
  for i := 1 to (t mod 5) do write('I');
```

```
нз
```

```
  if t = 4 then write('IV');
```

```
  if t < 4 then
```

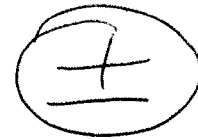
```
    for i := 1 to t do write('I');
```

```
нз
```

РАЗРЯД СОТЕН

ДЕСЯТКОВ

ЕДИНИЦЫ





N4.

Jf

Учитывая небольшое ограничение на N , написать решение
кватит на массив.

var

a: array [1..10000] of integer;

i, j, k, n: integer;

начало ввода (n);

a[1] := 4;

a[2] := 7;

a[3] := 10;

a[4] := 12;

i := 4; k := 1;

~~while i < n do~~

Пока (i < n) делать

нз

i := i + 1; a[i] := a[i-1] + 3;

for j := i + 1 to i + k do

a[j] := a[j-1] + 2;

i := i + k + 1; a[i] := a[i-1] + 3;

k := k + 1;

for j := i + 1 to i + k do

a[j] := a[j-1] + 2;

i := i + k;

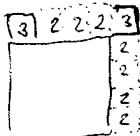
кц;

вывести (a[n]);

кцз.

~~Решение:~~

Пояснение:



Идея алгоритма в том, что поперечку
пополю «оборота» вокруг квадрата
робот тратит сначала 3 ребра, затем
к раз тратит 2, затем опять 3 и потом
k+1 раз 2 ребра. Соответственно k с
каждым разом увеличивается.

Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7113

ФАМИЛИЯ БАСАРГИНА

ИМЯ СВЕТЛАНА

ОТЧЕСТВО ЮРЬЕВНА

Дата рождения 19.07.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 5 листах

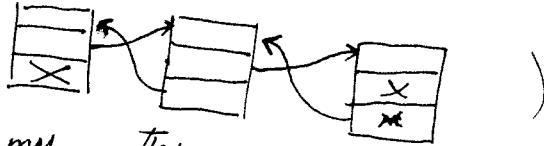
Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№1 Для перевода числа нам необходимо восстановить последовательность Фибоначчи до ^{заданного} числа N (для этого мы используем список, т.к. не знаем сколько будет переменных; в данном списке можно переходить как к предыдущему элементу так и к последующему)



Алгоритм Перевод;

Переменные ~~a, b~~ N: число; // N - заданное число; a и b необходимы для создания послед. т.к. по списку нельзя готовить определенные элементы.
a, b: натуральные;

Начало

ввод (N);

если (N < 0) то

вывод ("Значение числа неверно");

~~вывод~~

~~вывод~~

если (N = 0) или (N = 1) то вывод (N, "=", N);

если (N > 1) то

нц

создать список;

первый элемент списка = 1; a = 1;

создать элемент;

новый элемент списка = 2; b = 2;

пока (b < N)

нц

создать новый элемент;

новый элемент = a + b;

a = b;

b = новый элемент;

кц;

~~удалить~~ перейти к предыдущему элементу;
удалить последний элемент списка;

вывод (N, "=");

пока (существует список)

нц

если (элемент списка < N) то

= N = N - элемент списка;

вывод ("10");

удалить элемент;

перейти к предыдущему;

удалить элемент;

перейти к предыдущему

кц

иначе нц

вывод ("0");

удалить элемент списка; перейти к предыдущему

кц кц кц





16. Т.к. мы не знаем точно количество элементов, площади участка, значит нам необходимо создавать список элементов а не массив

Преимущество списка в том, что для него не обязательно знать кол-во элементов изначально, но "индекс", что в списке можно обратиться к определенному элементу, а необходимо двигаться от начала к концу.

Т.к. по условию заранее нам необходимо узнать кол-во элементов данного вида => раскломение данного элемента нам запоминать нет необходимости, а можно лишь идти по каждому семени и запоминать его вид.

Каждой элемент списка в нашем случае будет состоять из ~~двух~~ формы в которую входит 2 элемента: 1. - кол-во зерен (от 1 до N) и кол-во семян

Пример:

Список семян	
кол-во зерен	кол-во элементов
1	...
...	...
N	...
...	...

Обозначим:

список_семена $\{ (от\ 1\ -\ N) \}$

список_количество

Алгоритм Жосефа;

Переменные M, N, K, L, P: натуральные; f: логическая

C: натуральные // текущий посев;

Начало

ввод (M, N, K, L, P)

если (M < 0) или (N < 0) или (K < 0) или (L < 0) или (P < 0) то
вывод ("Некорректные параметры");

иначе ии

создать список

ввод (C);

первый элемент := список_семена $\{ C \}$;

первый элемент := список_количество = 1;

пока $(\exists \text{ элемент } c \in \text{список} \text{ с } f = \text{true})$ ии
от 1 элемента списка до последнего

ии если список_семена = C то

ии список_количество = список_количество + 1;

ии f = true;

если f = false то ии

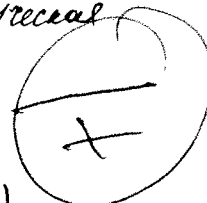
создать новый элемент

список_семена = C;

список_количество = 1

ии
ввод (C)

ии пока существует список ии
вывод (список_семена, список_количество);





№6 ~~удалить~~ элемент;
конец.

№5 Для решения данной задачи для начала можно все числа записать одномерной массивом состоящим из N-11 клеток, отсортировать его, а потом при выводе массива записать его в виде двумерного (MxN клеток); при этом ~~фрагмент~~ т.к. фрагмент массива для ~~не~~ знаем, запишем примерно

Алгоритм Сортировка;

Переменные O, N, M; ~~каждый элемент~~ $a[1..10^4]$ целые числа



Начало j; i: натуральное;

ввод (M, N);

если (M <= 0) или (N <= 0) то

вывод ("Некорректные данные");

иначе

для i = 1 до M*N

конец
ввод (a[i]);

конец
для i = 1 до M*N

конец
для j = 1 до M*N-1

конец
если a[j] > a[j+1] то

конец
~~temp := a[j];~~ o := a[j];

a[j] := a[j+1];

a[j+1] := o;

конец

конец
конец k = 0;

конец
для i = 1 до M*N делать

конец
вывод (a[i]);

k := k + 1

если k = M то

конец

вывод ("Enter");

k = 0;

конец

конец

№4 Алгоритм Стенки;

Переменные ~~...~~ // A = N^M

i - натуральное;

~~...~~ N, M, A := 1; ~~...~~

~~...~~

вывод (N, 1);





(N4) Т.к. число N возможно придется возводить в "большую" степень M , то лучше в этом случае воспользоваться алгоритмом быстрого возведения числа в степень.

Для этого для начала ~~возведем~~ переведем степень M в двоичный вид.

$$M_{10} = m_k m_{k-1} \dots m_1$$

По схеме Горнера: ~~(2m_k + m_{k+1})^2 + m_{k-2} \dots~~

$$N^M = N^{((\dots (2m_k + m_{k+1})^2 + m_{k-2} \dots) m_1)} = N^{2^{k-1} \cdot m_k} \cdot N^{2^{k-2} \cdot m_{k-1}} \cdot \dots \cdot N^{2 \cdot m_1}$$

Алгоритм степени;

Переменные k, i : натуральные;

A, N, M : целые; а массив $[1 \dots 10000]$;

начало

вывод ($N; M$); или

или ($N \leq 0$) ($M \leq 0$) то вывод (~~некорректное значение~~)

вывод ('Некорректное значение')

иначе

иц $i := 0$

пока $M > 2$

иц

$i := i + 1$

$a[i] := M \bmod 2$;

$M := M \operatorname{div} 2$;

иц

$i := i + 1$;

$a[i] := M$; $A := 1$;

для $k := i$ до 1 по убыванию

$A = A \cdot N^{(2^{k-1} \cdot a[k])}$;

иц

вывод (A);

иц
конец.

* если ($i > 1000$) или ($N \operatorname{div} 10^{1000} > 1$) то

вывод ('некорректное значение');

иначе

иц



(N2) Алгоритм Поиск;
 Переменные $N_1; N$: целые
 k : целое
 массив $a [1..50]$; min : натуральное
 Начало
 ввод (N_1).
 Если $N_1 < 100$ то вывод ("значение не корректно");
 создать список;
 для $k := 1$ до $N_1 + 50$
 иц
 создать новый элемент;
 новый элемент = $k * k$;
 ky ; $min = 100$;
 для $N := N_1 + 1$ до $N_1 + 50$
 иц
~~выбираем сумму квадратов из списка~~
~~пока не конец списка~~
 пока не конец списка
 иц
 подбираем сумму квадратов из списка;
 или сумма существует то
 $a[N-50] = N[N-50] + 1$;
 ky
 или $a[N-50] < min$ то $min = a[N-50]$;
 ky
 для $N := 1$ до 50
 иц
 или $a[N] = min$; то
 вывод ($N+50$, '-');
 ky
 конец.

Для решения данной задачи используется список состоящий из квадратов чисел от 1 до $N_1 + 50$

(7)

(N3) При выполнении алгоритма если заданное число p четное ($p \bmod 2 = 0$), то такое число сразу можно сказать, что составное. \Rightarrow по данному алгоритму можно рассматривать только нечетные числа.
 Число a должно выбираться через цикл от $z=1$ до \sqrt{p}
 a - любое число ($Random(p-1)$)
 после определения a должно осуществляться проверка числа $p-1 = 2^s \cdot d$ по данному методу
 где оптимальным заданием необходимо сначала проверить простое ли число с $n.1 (a^d = 1 \bmod p)$ и если эта проверка прошла то через цикл от 0 до $s-1$ проверить условие 2. Если и это условие можно вывести "составное"

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Большаков

ИМЯ Амтрий

ОТЧЕСТВО Валерьевич

Дата рождения 07.08.1994

Класс: 11

Предмет Информатика

Этап: Заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 24.02.2015
(число, месяц, год)

Подпись участника олимпиады: *Большаков*

пишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.

Задача 1

время 1 доп. мес

с/л

Начало

Ввод (n);

R[1] := 1000; RN[1] := 'M';

R[2] := 500; RN[2] := 'D';

R[3] := 100; RN[3] := 'C';

R[4] := 50; RN[4] := 'h';

R[5] := 10; RN[5] := 'X';

R[6] := 5; RN[6] := 'V';

R[7] := 1; RN[7] := 'I';

i := 1;

Пока n > 0 делай

Начало

Пока n div R[i] = 0 делай i := i + 1;P := n div R[i];

если P ≤ 3 тогда

иначе для j от 1 до P делай s := s + RN[j];

начало

для j от P до 4 делай s := s + RN[j+1];

s := s + RN[i];

конец;

n := n mod R[i];

конец;

Вывод (s);

Конец.

n - исходное натуральное число;

P, L, i, S, RN[1..7] - целые числа;

RN[1..7] - шифровальные буквы;

S - строка;



Задача 2

Очевидно, что в 2-10 системе на число отводится не меньше (чаще всего - больше) информации, чем в 2 системе. Но, не смотря на это, у 2-10 системы есть своя ниша: ~~действительные~~^{мощные} ~~основанные~~ на действиях с отдельными цифрами десятичной записи, например - их перестановкой. Также 2-10 система более удобна и понятна человеку, привыкшему считать в 10 системе. (+)

Задача 3

Начало
Ввод (n, a);
 $c := \text{истина}$;
 $v := 2$.

c - логическая пер-мая;
 n - модуль;
 a - целое число;
 v, i - целые числа;

Пока (c) и $(v < n-1)$ делай

Начало
Пока $((a \cdot v) \bmod n < > 1)$ и $(v < n-1)$ делай $v := v+1$;

Если $(a \cdot v) \bmod n = 1$ то начало $c := \text{ложь}$;

для i от 2 до \sqrt{a} делай

Начало
если $(a \bmod i = 0)$ и $(v \bmod i = 0)$ то
 $c := \text{истина}$;

Конец;

Конец;

если не (c) и $((a \cdot v) \bmod n = 1)$ то вывод(v)
иначе вывод ('не существует').

Конец.



Задача 4

Идя по спирали, на первую клетку пойдет чрепки, на каждый поворот — 3 рейки, на остальных клетках — по 2 рейки. Назовём участок «прямым», если его ограничивают повороты, а сам участок состоит только из 2-речных квадратов. По прямой спирали, длины таких прямых участков будут увеличиваться через один, но есть сначала идут 2 участка длины a , затем 2 участка длины $a+1$ и т.д. Каким образом, для некоторой длины N на реке можно пройти все квадраты, руководствуясь следующими выше правилами

Начало

Ввод(N); $S := 0$; $i := 1$; $L := 2$; $K := 1$;Пока $N > 0$ делай

Начало

Если $N = 1$ то начало $S := S + 4$; $N := 0$; конец;Если $N = 2$ то начало $S := S + 6$; $N := 0$; конец;Если $N = 3$ то начало $S := S + 10$; $N := 0$; конец;Если $N > 3$ то

Начало

 $N := N - 1$; $S := S + 2$;если $L = 0$ то начало $L := 2$; $K := K + 1$; $i := K$; $S := S + 1$; конец;если $i = 0$ то начало $L := L - 1$; $i := K$; $S := S + 1$; конец; $i := i - 1$;

конец;

Вывод(S);

Конец.





djf

Задача 5.

Вводя данные, мы заполним или три ~~или~~ одинаковых массива, соответствующих столбцам таблицы.

Возьмем первый номер и поместим его в строку. Затем возьмем второй номер и поместим ему место в строке, сравнив ~~первый~~ с первым: если второй больше — то разместим его правее, иначе — левее, оставшаяся пробела между номерами. Аналогично будем помещать в строку остальные номера, пропуская строку (или строки, если номеров много) слева направо.

После окончания заполнения мы по тому порядку номеров, в котором они в строке/строках, мы заполним новые три массива. Таким образом мы получим упорядоченные данные по возрастанию результата.

Задача 6

Пусть школьники пишут по одной или несколько цифр в клетке, разбивая числа по столбцам (т.е. по столбцам), записывая их в строку, а при окончании одной — переходя на новую. Тогда мы будем аналогично считывать: сразу по строке из M символов. Числа будем считывать в столбцовом формате, собирая в одну кучу между пробелами. Каждый раз, встречая окончание числа (т.е. пробел, до которого нечетная клетка), мы будем принимать перевернутое считанное так, что всегда будем знать номер партии или результат мы считали. Если это результат — мы прибавим его к общей сумме соответствующего игрока. После всех подсчетов мы определим победителя, сравнив общие суммы мальчиков. (+)

Олимпиада школьников «Надежда энергетики»

[Empty box for group number]

№ группы

Вариант № 7092

2E 51-80

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ВАСИЛЬЕВА

ИМЯ Анна

ОТЧЕСТВО ОЛЕГОВНА

Дата рождения 26.10.1999

Класс: 9

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 2 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



1. В позиционной системе счисления цифра, стоящая в определенном разряде не может быть больше q ; здесь q — основание не является определенным числом, это зависит от разряда

1. Цифра, стоящая в определенном разряде, относится на определенное число, которое постоянно для данного номера разряда во всех числах. Каждая цифра занимает свое свое место (разряд). Одна и та же цифра может быть в любом разряде. Значит эта с.с. является позиционной
 Ответ: да, является

2. Данные надо хранить в таблице:

Страна	основное время	Переводят ли часы	Когда переводят (от и до)	Переводят вперед или назад
--------	----------------	-------------------	---------------------------	----------------------------

В 1 столбце: название страны

В 2 столбце: время без перевода часов (например, GMT+1)

В 3 столбце: $+1$ или 0 переводят ли часы в данной стране (1, если переводят; 0, если не переводят)

В 4 столбце: период, когда часы переведены (от какого-то числа до какого-то числа), т.е. время отнимается от основного, которое представлено во 2-ом столбце

В 5 столбце: в какую сторону переводят время (вперед — $+1$ час; назад — -1 час)

Если часы не переводятся в данной стране, то 4 и 5 столбцы остаются пустыми

Чтобы определить время нужно:

1) Найти нужную страну в 1-ом столбце

2) Узнать, переводят ли время в этой стране (в 3 столбце):

- если переводят, то узнать, когда переводят (4 столбце):
 - если сегодняшняя дата входит в данный период, то узнать, в какую сторону переведено время (5 столбце):
 - ▲ если $+1$, то к основному времени прибавить 1, получится искомое время
 - ▲ если -1 , то из основного времени (2 столбце) вычесть 1, получится искомое время
 - если сегодняшняя дата не входит в данный период, то время, стоящее во 2 столбце — искомое.
- если не переводят, то время из 2 столбца — искомое

3.

• R присвоить $p-1$;

• K присвоить 0;

• Выпалывать циклы, пока $R > 0$:

1) T присвоить R; B присвоить P;

2) Выпалывать циклы, пока T не равно B: если $T > B$, то T присвоить $T-B$, иначе B присвоить $B-T$;

3) T присвоить B, если $T=1$, то K присвоить $K+1$;

4) R присвоить $R-1$;

• K — искомое



4.

- K присвоить 1;
- Прогнать M раз: K присвоить K.N; \ominus
- K - искомое число (NM)

5. $4 \cdot 11 = 44$

- K присвоить 0;
- Выпалить циклы, пока $K:10000 < 1$: K присвоить $K+44$; \oplus
- ~~Выпалить K;~~

- Выпалить циклы, пока $K:100.000.000$: выпалить K; K присвоить $K+44$

6. Если число будет максимально близко к нулю, то компьютер может его округлить до нуля ($0+1=1$) \ominus

7. K присвоить 44; S присвоить 1;

- Выпалить 44 раза: S присвоить $S \cdot K$; K присвоить $K-1$;
- T присвоить S; L присвоить 0;
- Выпалить циклы, пока $T > 0$: T присвоить целую часть от деления T на 10 ($T \div 10$); ~~X присвоить $L-1$~~ ; L присвоить $L+1$;
- X присвоить $L-1$
- Выпалить циклы $L-1$ раз: N присвоить целую часть от деления $S \cdot X$ на $10 \cdot X$ ($S \div (10 \cdot X)$); если $N=4$, то записать $X+1$; S присвоить остаток от деления S на $10 \cdot X$ ($S \bmod (10 \cdot X)$); X присвоить $X-1$;
- Если $S=4$, то записать 1. \ominus

Выдан документальной мсм
Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7102

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Вахрушев

ИМЯ Кирилл

ОТЧЕСТВО Константинович

Дата рождения 10.06.1998

Класс: 10

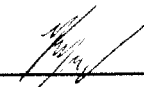
Предмет информатика

Этап: заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



1. $X_8 \rightarrow X_{10} \rightarrow X_{\text{десятич. с. с.}}$

Пример на Паскале:

```
var i, n, s, p, j, k: longint
```

```
begin
```

```
  read(n) // считываем число в 8-ой с. с.
```

```
  s := 0; // здесь храниться (будет) десятичная запись числа
```

```
  n := n * 10;
```

```
  j := 0; // счетчик степеней при переводе в десятич. систему
```

```
  for i := 1 to (n div 10) do // n div 10 - кол-во цифр в исходном числе
  begin
```

```
    p := n mod 10; // выбираем последнюю цифру
```

```
    n := n div 10; // убираем из числа последнюю цифру
```

```
    s := s + p * 10j; // перевод цифр разряд в десятич. с. с. и добав. к сумме
```

```
    j := j + 1; // увеличиваем счетчик степени
  end;
```

// Теперь в "s" у нас имеется исходное число в десятичной записи

```
k := s div 10; // увеличимся дальше на 10
```

```
s := s mod 10; // оставляем только последний разряд < 10;
```

```
for i := 1 to k do write('x') // выводим все десятки
```

```
if s = 1 then write(', I'); if s = 6 then write(', VI');
```

```
if s = 2 then write(', II'); if s = 7 then write(', VII');
```

```
if s = 3 then write(', III'); if s = 8 then write(', VIII');
```

```
if s = 4 then write(', IV'); if s = 9 then write(', IX');
```

```
if s = 5 then write(', V');
```

```
end.
```

Выводим последнюю цифру.

2. Дважды-десятичную систему счисления целесообразно использовать, когда мы работаем с дробной записью числа, а выводим результат в виде десятичной системы счисления. Например, нам надо представить цифры в числе:

21 → 12

00100001 → 00010010

Это возможно т.к. мы точно знаем размер (4 бита) десятичной цифры в двоичном представлении.



7. Пример на Паскале:

$d1$ - длина реки текущая сверху вниз, длина боковой стороны

sk - сколько квадратов построено

sv - кол-во задействованных рек



```
var d1, sk, sv, n : longint
```

```
begin
```

```
read(n) // считываем сколько нужно построить квадратов
```

```
if n=1 then write(4) else
```

```
if n=2 then write(7) else
```

```
if n=3 then write(10) else
```

```
if n=4 then write(12) else
```

```
begin
```

```
  d1:=2;
```

```
  sk:=4;
```

```
  sv:=12;
```



```
while sk < n do
```

```
begin
```

```
  sk := d1 * 4 + 4
```

// квадр. будем построено столько же, сколько в предыдущем слое + квадраты на углах

```
  sv := d1 * 4 * 2 + 3 * 4
```

// для построения четырех угловых нужно 3 реки
а для остальных 2 реки

```
  d1 := d1 + 2
```

// сторона боковая с каждым слоем увеличивается на 2

```
end;
```

```
write(sv); // кол-во задейств. рек, вывод
```

```
end.
```



5.

Данные будем удобно хранить в массиве с указателями(a)

$a[n], n$

↑
 n^o/n

$a[i]$ object

↑
string

- значение параметра

$a[n].result$ - результат

↑
longint

Сортировку будем осуществлять - сортировкой пузырьковой,



но если данных очень много, то можно воспользоваться и быстрой сортировкой (quicksort)

Будем считать, что у нас в массив a записаны все данные тогда пример сортировки будет выглядеть так:

```

for j := 1 to n-1 do // n - кол-во значений
  for i := 1 to n-j do
    begin
      if a[i].result > a[i+1].result then begin
        z := a[i]
        a[i] = a[i+1]
        a[i+1] = z
      end;
    end
  end

```

(7)

Все массив, т.е. таблица, отсортирована по убыванию, задача решена.

6.

Будем считать, что вся информация нам дана в виде двумерного массива с M столбцами и N строками ($a.array[1..M, 1..N]$ of T)

ПАСКАЛЬ

$K := 1$ // номер столбца „№ партии“

$P := 0$;

$D := 0$; // Обнуляем кол-во очков у ПЕТРА и ДАНИЛА

for i := 1 to $M \div 3$ do // $M \div 3$ - это кол-во таблиц в столбце, делим по 3

for j := 1 to N do // пробегаем по всем строкам

$P := P + a[K+1, j]$; // прибавим к очкам его столбца (у Петра)

$D := D + a[K+2, j]$; // аналогично для Даниила

$K := K + 3$; // меняем указатель на следующую ^{столбец} „№ партии“

end;

if $P < D$ then write („Petra“) else

if $D < P$ then write („Davit“) else write („Ничья“);

end.

3. ↓
задание



3. Пример на Паскале.

```
var n, i, a, b, n1, a1, m1: longint;  
begin  
  read(n, a);  
  k:=1; // указатель на минимальный  
  m1:=n;  
  a1:=a; // создаем дубликат a и n  
  while a1 <> m1 do  
    if a1 > m1 then a1:=a1-m1 else m1:=m1-a1; // находим НОД  
  if m1 <> 1 then write("не существует") else  
    for b:=1 to n do // перебором находим b, удовлетворяющее  
      if (a-b) mod n=0 then // наименьшее условию (доп.т.к. b < n)  
        begin  
          write(b); // выводим подходящее число  
          k:=b; // указатель - найден  
        end  
  if k=0 then write("не существует") // если не нашли.  
end.
```



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ВОЛКОВ

ИМЯ ГРИГОРИЙ

ОТЧЕСТВО ВАЛЕРЬЕВИЧ

Дата рождения 05.10.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 7 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ЛЧ.

виды 1 гол. лист + 1 гол. лист + 1 гол. лист + 1 гол. лист
ЛЧ

Заметим, что 3 внешних ребра добавляются для квадрата, которые ставятся сразу после того, как мы доиграли 1 боковую. Иными словами для N вида $k^2 + 1$ ($k \in \mathbb{N}$ натур. числа). Также 3 ребра можно для квадрата, который выхо-
дит за границу прямоугольника. Иными словами вида $k^2 + k + 1$ (k - натур. число). $N \leq 1$ - обра-
батываем отдельно. Тогда невоиную (Решая)
будет выглядеть так:

```

if (N=1) then writeln ('4');
if (N <= 1) then if ((trunc(N))^2 + 1 = N)
or ((trunc(N))^2 + 1 = N)
if (N <= 1) and (((trunc(sqrt(N)))^2 + 1 = N)
or ((trunc(sqrt(N)))^2 + trunc(sqrt(N)) + 1 = N))
then writeln ('3') else writeln ('2');
```

Здесь trunc - ф-ия, которая отбрасывает дроб-
часть, sqrt - ф-ия, которая берет квадрат-
корень. Увеличивая то, что возводим в квадрат
намши отбрасыв.



З3.

1) Сначала описали функцию $\text{nod}(x, y)$, которая будет считать НОД(~~x, y~~) (для n -элементов).

2) Потом перебор от 1 до $(n-1)$ чтобы найти число b , чтобы выполнялось $(a \cdot b) \bmod n = 1$.

3) На псевдокоде (Pascal) алгоритм будет выглядеть так:

1) Описание функции nod :

```
function nod (x, y: longint): longint;
```

```
begin
```

```
while (x <> 0) and (y <> 0) do
```

```
if (x > y) then x := x mod y else
```

```
y := y mod x;
```

```
nod := x + y;
```

```
end;
```

2) Сам алгоритм:

```
if (nod(a, n) = 1) then for i := 1 to n do
```

```
if (i * a) mod n = 1
```

```
then writeln(i);
```

```
if (nod(a, n) <> 1) then writeln('He exists');
```



25.

1) Пусть у нас есть 3 лямбда-массива:

$a[i]$ - № и.и. (число)

$b[i]$ - значение параметра (строка)

$c[i]$ - результат (число)

а также n - кол-во элементов.

2) В плохом коде ~~здесь~~ ^{рекурсивно} сортировка, которую
нашему «куратору» на собеседовании (Pascal):

```
for i:=2 to n do
  for j:=2 to n do
```

```
  begin
```

```
    if (c[j-1] > c[j])
```

```
  then begin
```

```
    t:=c[j];
```

```
    c[j]:=c[j-1];
```

```
    c[j-1]:=t;
```

```
    t:=a[j];
```

```
    a[j]:=a[j-1];
```

```
    a[j-1]:=t;
```

```
    w:=b[j];
```

```
    b[j]:=b[j-1];
```

```
    b[j-1]:=w;
```

```
  end;
```

```
end;
```

Примечание! Максим значение элементов массива
методом «брезвей параметра», t - числовая, w - строка



16. 1) Неизвестно, как вводится значение. Допустим, мы считаем как дается кол-во карт (N), а потом N строчек следующего вида:

$$k \quad a \quad b$$

где k - номер карты, a - кол-во очков Девы, b - кол-во очков Дамы. Короче говоря, у нас есть 2 ящики S_1, S_2 (числовые), в которых мы храним сумму очков Девы и Дамы.

2). На языке Pascal можно написать алгоритм будет примерно так:

```

readln (n);
s1 := 0; s2 := 0;
for i := 1 to n do
begin
  readln (k, a, b);
  s1 := s1 + a;
  s2 := s2 + b;
end;
if (s1 > s2) then writeln ('Девы');
if (s2 > s1) then writeln ('Дамы');
if (s1 = s2) then writeln ('Ничья');

```





51.

mf

Алгоритм нужно выполнить на число (целое введенное число - N):

1) Определяем кол-во букв 'M' где введена, как $(N \text{ div } 1000)$, затем $N := N \text{ mod } 1000$.

2) Определяем, как как введем разряд согласно следующ. алгоритму:

а) если $(N \text{ div } 100) = 4$, то выведем 'CD';
 $N := N \text{ mod } 100$;

б) если $(N \text{ div } 100) = 9$, то выведем 'CM';
 $N := N \text{ mod } 100$;

в) если $(N \text{ div } 100) \in [1; 3]$, то соответствующее кол-во 'C'

г) если $(N \text{ div } 100) \in [5; 8]$, то выведем 'D'
и $(N \text{ div } 100 - 5)$ раз выведем 'C'

3) Аналогично для разряда десятков, только:

а) вместо $(N \text{ div } 100)$ пишем $(N \text{ div } 10)$

б) в пункте: а) выведем 'XL'

б) выведем 'XC'

в) соотв. кол-во 'X'

г) выведем 'L' и $(N \text{ div } 10 - 5)$ раз 'X'

4) То же самое для единиц, только:

а) вместо $(N \text{ div } 100)$ пишем N

б) в пункте: а) выведем 'IV'

б) выведем 'IX'

в) соотв. кол-во 'I'

г) выведем 'V' и $(N - 5)$ раз 'I'

На следующ. странице алгоритм на псевдокоде:



Программа №1 (Плюсатор):

```
readln (N);
for i:=1 to (N div 1000) do
  write ('M');
N:= N mod 1000;
if (N div 100 = 4) then write ('CD');
if (N div 100 = 9) then write ('CM');
if ((N div 100) in [1..3]) then for i:=1 to (N div 100) do
  write ('C');
if ((N div 100) in [5..8]) then for begin
  write ('D');
  for i:=1 to (N div 100 - 5) do
    write ('C');
  end;
N:= N mod 100;
if (N div 10 = 4) then write ('XL');
if (N div 10 = 9) then write ('XC');
if ((N div 10) in [1..3]) then for i:=1 to (N div 10) do
  write ('X');
if ((N div 10) in [5..8]) then begin
  write ('L');
  for i:=1 to (N div 10 - 5) do
    write ('X');
  end;
N:= N mod 10;
if (N = 4) then write ('IV');
if (N = 9) then write ('IX');
if (N in [1..3]) then for i:=1 to N do
  write ('I');
if (N in [5..8]) then for i:=1 to (N-5) do
  write begin
  write ('V');
  for i:=1 to (N-5) do
    write ('I');
  end;
```



Лонг

№2.

1) Задача сводится к следующему вопросу:
нужно найти такое число N , чтобы оно было
выборочно хранится в 2-10 системах счисления,
чем в 2.

2) а). Пусть k - кол-во цифр искомого числа N ,
тогда запись его в сист. счел. 2-10 будет
записана $4k$ бит

б). Пусть x - максимальное возможное число,
тако что $2^x \leq N$, тогда в 2 системе счисления
будет запись $(x+1)$ бит.

3). Тогда требуется найти такое N , чтобы
 $4k < (x+1)$.

4). Рассмотрим ~~k от 1 до 4~~ различные k и при

них ~~находим~~ ~~максимум~~ x : максимизируем x :

$$k=1 \Rightarrow \max x = 3 \Rightarrow 4 < 4 \quad (\text{неверно})$$

$$k=2 \Rightarrow \max x = 6 \Rightarrow 8 < 7 \quad (\text{неверно})$$

$$k=3 \Rightarrow \max x = 9 \Rightarrow 12 < 10 \quad (\text{неверно})$$

$$k=4 \Rightarrow \max x = 13 \Rightarrow 16 < 14 \quad (\text{неверно})$$

$$k=5 \Rightarrow \max x = 16 \Rightarrow 20 < 17 \quad (\text{неверно})$$

$$k=6 \Rightarrow \max x = 19 \Rightarrow 24 < 20 \quad (\text{неверно})$$

Очевидно, что и при большем k разность между
 $x+1$ и $4k$ будет только расти. Скорее всего,
2-10 системы счисления не выборочно ~~не~~ но только
для двух, но и для раз.ых чисел. Единственный
её плюс только в том, что ~~не~~ с ней проще про-
изводить операции перевода в др. системы счисления.

(7)
+

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ГОНЧАРЕНКО

ИМЯ Дмитрий

ОТЧЕСТВО АЛЕКСАНДРОВИЧ

Дата рождения 03.11.1997

Класс: 11

Предмет Информатика

Этап: Заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 15.03.15
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



□

Алгоритм перевода $X_{10} \rightarrow Y_{\text{Fib}}$:Введем ~~два~~ целочисленные переменные p и n ;Запишем число, которое требуется перевести в переменную n ;Создадим массив V из n элементов ($\text{int } V[n]$);Для первой и второй элементу массива присвоим значения 1 и 2 соответственно. ($V[0]=1, V[1]=2$);Введем переменную $k=1$ Пока k ~~не~~ элемент в массиве $k \leq n$ ($V[k] < n$)нужно { $V[k+1] = V[k] + V[k-1]$; (находим следующее число Фибоначчи)Увеличиваем k на единицу ($++k$); k переменной p присваиваем значение k ($p=k$);

конец }

Введем массив u из p элементов $\text{int } u[p]$;Заполним этот массив нулями $V[1..p] = \{0, 0, \dots, 0\}$ Присвоим k значение $p-1$ ($k=p-1$);Пока $k \geq 0$ нужно { $n = n - V[k]$; // присвоим k разность n и ближайшего меньшего числа Фибоначчи $u[k] = 1$;// i элементу массива u присвоим единицу, что соответствует позиции числа Фибоначчи, которое мы выли в предыдущем действии.Если n не равняется 0, тоПока ($V[k] > n$) { $--k$; }Иначе $i = -1$; // выйдем из цикла

конец }

Выводим массив u начиная с последнего элемента (получим число Фибоначчи)



4

Ввод M и N .

Пока $M > 0$

нч $N = N * N;$

$M = M - 1;$



кч

Вывод $N;$

5

Ввод M и $N;$

Создадим массив $V [M * N]$ (одна строка)

Цикл k от 0 до $M * N$

// for (int k=0; k<M*N, ++k)
for (int i=k, i<M*N, ++i)

нч Циклом k до $M * N$

нч ~~if~~ если $(V[i] < V[k])$, то меняем их местами ~~swap(V[i], V[k])~~

кч

кч



Цикл k от 0 до $M * N$ нч // столбцы

Цикл i от 0 до M // строки

нч (Выводим значения массива V) V
переходим на следующий столбец. $(++M)$

кч Переходим на следующую строку $(++N)$

кч.

// В результате получим таблицу M на N заполненную убывающими числами.



3

Ввод: цел: $p \geq 2$; $k = p - 1$, $s = 0$; $t = \log_2(p)$.
 bool: test = true.

Пока (k : на 2)
 ну k делит на 2 ($k = k / 2$)
 s увеличивается на единицу ($++s$)
 к у

$$d = \frac{p-1}{2^s};$$

$k = t$;
 Пока $k > 0$

/* Проверка 1 */
 ну Выбери случайное число a от 1 до $p-1$
 Если a^d не равняется 1 по модулю p , то test = false.
 -- k ; (уменьшаем k на единицу)
 к у

/* Проверка 2 */ Цикл от $r = 0$ до $r = s - 1$

ну Пока $t > 0$

ну выбери случайное число a от 1 до $p-1$

введи цел $m = d \cdot 2^r$;

Если a^m не равняется -1 по модулю p , то test = false

-- t ; (уменьшаем t на единицу)

к у

Если test == false, то вывод "составное"
 Иначе вывод "вероятно простое".





гитис

ST 8T 88

№ 6

Для решения этой задачи нам потребуются списки.

Создадим список строк. (т.к. поле прямоугольное)

Создадим список столбцов

Добавим элемент в список строк и сделаем его текущим.

(первый элемент списка строк должен соответствовать первой строке в которой есть хотя бы одна ель) (т.к. расположение саженцев неизвестно, будем искать ель, на, оцирк)

(для этого требуется разбить поле на квадраты по 2 м^2)

Пока $(P > 0)$ (все ели найдены) $\& \& \&$ (\oplus)
ну $(\text{площадь закончилась})$
введем переменные $P = P - 1$; $S = 1000k * 1000l$, $k = 0$, $i = 1$, $n = 1$ (от 1 до n)

Пока в строке есть ель
Добавим в список ~~строки~~ ^{столбцовый} элемент и сделаем его текущим

Запишем кол-во ярусов у этой ели j ; $P = P - 1$; $S = S - z$; $++i$;

Добавим новый элемент в список строк и сделаем его текущим
Запишем кол-во элементов в строке в текущий элемент списка. (k_n)
 $++n$

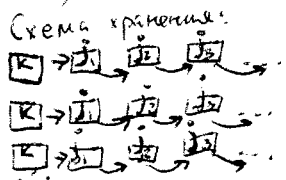
Установим текущим элементом списка строк первый элемент в строке.
Перейдем на первый элемент списка столбцов. $n = 1$;

Создадим массив $V [M]$ элементов. и закончим его нулями.
Пока существует следующий элемент списка строк. ну.

Пока в ~~строке~~ ^{текущем} элементе списка строк $k_n > 0$
ну $V[j] = V[j] + 1$;
перейдем на следующий элемент в списке столбцов
 $++i$; $--k_n$
 $i = 1$; $++n$;

Выводим массив V , начиная с элемента M .

(Получим кол-во елей каждого вида)
Выводим $P - P$ (сколько саженцев не возшло)



Олимпиада школьников «Надежда энергетики»

[Empty box for group number]

№ группы

Вариант № 7102

С W 59 - 96

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ГРИГОРЬЕВ

ИМЯ ДМИТРИЙ

ОТЧЕСТВО ВЛАДИМИРОВИЧ

Дата рождения 26.06.1998

Класс: 10

Предмет информатика

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 8 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:

[Handwritten signature]

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



N4. Выдан 1 доп. лист. Выдан 2^{ой} доп. лист Выдан 3 доп. лист
Лисенко Ван Лисенко

В этой задаче я нашел закономерность:

1) 4 рейки понадобятся только в одном случае Выдан доп. лист N4
Лисенко
Выдан доп. лист N5
Лисенко
для одного квадрата.

2) 3 рейки понадобятся в том случае если Лисенко
 $(n-1)$ -какой квадрат целого числа.

$(n-1)$ равно (целой части от корня из n) умно-
женной на (целую часть корня из $n+1$).

3) 2 рейки во всех остальных случаях.

Решение к этой задаче я сделал на языке Free Pascal.

```

program z4;
var n: integer; i, s: longint;
begin
  readln(n);
  if (n=1) then writeln („Понадобится 4 рейки“) else
  begin
    s:=4;
    for j:=2 to n do
      if (trunc(sqrt(n-1))=sqrt(n-1) or (trunc(sqrt(n)) * trunc(sqrt(n)+1)=
        =(n-1)) then s:=s+3 else s:=s+2;
    writeln („Понадобится“, „ „, s, „ „, „реек“);
  end;
end.

```

Получилось, что в s всегда находится минимальное количество реек.



N 1

Я буду решать эту задачу следующим образом: составлю две функции, одна из которых переводит число в восьмеричной в десятичную, а вторая из десятичной в римскую. Решение сделаю на языке Free Pascal:

```
program z1;  
var
```

```
function iz8v10 (n:longint):longint;  
var i,j,step,ch,ans,x:longint;  
s:string;
```

```
begin  
str(n,s); {переводим число в строку s}  
ans:=0;  
for i:=0 to length(s)-1 do  
begin  
step:=1;  
for j:=1 to i do  
step:=step*8;  
val(s[i],ch,x)  
val(s[length(s)-i],ch,x);  
ans:=ans+ch*step;  
end;  
iz8v10:=ans;  
end;
```

```
function izRim(n:longint):string;  
var s:string;  
i:longint;
```

```
begin  
for s:='';  
for i:=1 to n div 1000 do  
s:=s+'M';  
n:=n mod 1000;
```




```
if n div 100 = 4 then s := s + 'CD' else
  if n div 100 = 9 then s := s + 'CM' else
    if n div 100 = 5 then s := s + 'D' else
      if n div 100 > 5 then
        begin
          s := s + 'D';
          for j := 1 to (n div 100 - 5) do
            s := s + 'C';
          end
        else
          for j := 1 to n div 100 do
            s := s + 'C';
```

n := n mod 100;

```
if n div 10 = 9 then s := s + 'I' else
```

```
if n div 10 = 5 then s := s + 'L' else
```

```
if n div 10 = 4 then s := s + 'XL' else
```

```
if n div 10 > 5 then
```

```
begin
```

```
s := s + 'L';
```

```
for j := 1 to (n div 10 - 5) do
```

```
s := s + 'X';
```

```
end
```

```
else
```

```
for j := 1 to n div 10 do
```

```
s := s + 'X';
```

n := n mod 10;

```
if n div n = 4 then s := s + 'IV' else if n = 5 then s := s + 'V' else
```

```
if n = 9 then s := s + 'IX' else if n > 5 then
```

```
begin
```

```
s := s + 'V';
```

```
for j := 1 to n - 5 do
```

```
s := s + 'I';
```

```
else
```

```
begin
```

```
for j := 1 to n do
```

```
s := s + 'I';
```

```
if Rim := s;
end;
```



Выход 1 год. мск
Левар

```
begin
  Readln (n);
  if n <= 0 then writeln ("Невозможно
                           перевести") else
```

```
begin
  writeln (izRim (izBv10 (n)));
end;
end.
```

N3

Составить программу на языке Free Pascal:

```
program z3;
var n, a, b, i: longint;

function NOD (n, m: longint): longint;
var min, i, x: longint;
begin
  if n > m then min := m else min := n;
  x := 1;
  for i := 1 to min do
    if (n mod i = 0) and (m mod i = 0) then x := i;
  NOD := x;
end;

begin
  Readln (n, a);
```

этот лист №2
БП

```
if NOD (a,n) <> 1 then writeln("Не существует") else
begin
b := 0;
while (a * b) mod n <> 1 do
inc (b);
writeln (b);
end;
end.
```



N6.

В условии сказано как разложились входные данные, поэтому я буду считать, что в одной строке было три числа (N партий, rez1, rez2). Сказано, что размер страницы $M \times N$, значит всего было сыграно M партий.

Решение напишу на языке Free Pascal:

program z6;

var Petr, PAV, N, M, i: longint; p, rez1, rez2: integer;

begin

readln (M, N);

Petr := 0; PAV := 0;

for i := 1 to M do

begin

readln (p, rez1, rez2);

Petr := Petr + rez1;

PAV := PAV + rez2;

end;





Выдан доп. лист №3
Денис

```
if Dan = Petr then writeln ("Низь") else
  if Dan > Petr then writeln ("Петр") else writeln
    writeln ("Данил");
end.
```

№5

Ввод я буду производить построчно, т.е. в одну строку ввожу и N и P , значение параметра и результат. Хранить это буду в 2 массивах. Один массив - массив строк, в нем будет храниться N и P и значение параметра, в другом результат;

Для сортировки ~~буду~~ буду использовать алгоритм быстрой сортировки (quicksort). Он рекурсивный, поэтому для него напишу ~~сортировку~~ отдельную процедуру.

Писать буду на Free Pascal:

```
program z5;
var n, xi, j: longint; s, s1: string;
    mas: array [1..100000] of string;
    mas1: array [1..100000] of longint;

procedure quicksort (l, r: longint);
var i, j, x, temp: longint; tempt: string;
begin
  if l < r then begin
    i := l;
    j := r;
    x := mas1 [(i+j) div 2];
```





Выдан гол. лист №4
Пенюк

```
Repeat
  While mas1[i] < x do
    inc(i);
  While mas1[j] > x do
    dec(j);
  If i < j then begin temp := mas1[i]; temp1 := mas1[j];
    mas1[i] := mas1[j]; mas1[j] := temp;
    mas1[j] := temp; mas1[i] := temp1;
    inc(i); dec(j);
  end;
```

```
Until i < j;
quicksort(i, r);
quicksort(l, j);
end;
```

```
begin
  n := 0;
  While not eof do { пока не конец файла }
```

```
  begin
    readln(s);
    s1 := ""; s2 := "";
    j := length(s);
    while s[j] <> ' ' do
```

{ символ обозначает пробел }

```
  begin
    s1 := s[j] + s1;
    dec(j);
  end;
  for i := 1 to j - 1 do
    s2 := s2 + s[i];
    mas[n] := s2;
    val m(s1, k, c);
```



Выход док. лист №5
Лисенко

```
mas[1..n] := k;  
end;  
quicksort(1, n);  
for i := 1 to n do  
  write ln (mas[i], ' ', mas[1..i]);  
end.
```

Вывод можно убрать, т.к. в условии задачи не сказано вывести результат или нет. Алгоритм сортировки, который я применил является одним из самых быстрых алгоритмов сортировки из ныне существующих. Он работает за $O(n \log_2 n)$.

Поэтому сортировать программа будет быстро. гораздо выгоднее по памяти искать зовошь динамической массива, чтобы меньше занимать память, но это как с ними работать.

№ 2.

Применение двоично-десятичной системы счисления целесообразно в тех случаях, когда мы работаем только с числами. Т.е. без букв, символов, цитирований и т.д.

Это удобно тем, что перевод в двоичную систему счисления легко осуществляется с помощью триад и тетрад. И легко выполнять операции.

Её удобно использовать на устройствах, где надо вводить и выводить числа. Таковыми являются: кувшмятор, электронные часы, автомобильный определитель номера и цифровые ~~и~~ измерительные приборы (например цифровые термометры).

Олимпиада школьников «Надежда энергетики»

□

№ группы

Вариант № 7102

CW 59-38

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ИСМАИЛОВ

ИМЯ РАШАД

ОТЧЕСТВО МАХИР ОГЛЫ

Дата рождения 29.07.1998

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 6 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:

Исмаилов

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



~~Аналогичными названиями~~

Если $d_2=9$ и $d_3=9$, ~~то $x_2=$~~ и $d_1=9$, то $x_2=IM$

Если $d_2=9$ и $d_3=9$, но $d_1 \neq 9$, ~~то $x_2=$~~ и $d_4 > 4$, то

$$\cancel{x_2 = \overline{I D C C}}_{d_4=5} \quad x_2 = \overline{D \underbrace{C \dots C}_{d_1=5} I C}$$

Если $d_2=9$ и $d_3=9$ и $d_4=4$, то $x_2=ID$.

Если $d_2=9$ и $d_3=9$ и $d_1 < 4$, то $x_2 = \overline{I \underbrace{C \dots C}_{d_1} C I C}$.

Если $d_2=9$

Для всех остальных случаев ~~x_2~~ x_2 записывается как строка из цифр d_1, d_2 и d_3 в римской с.с.

Если $d_1 < 4$, то d_1 в рим.с.с. = ~~$C \dots C$~~ $\underbrace{C \dots C}_{d_1}$

Если $d_2 < 4$, то d_2 в рим.с.с. = $\underbrace{X \dots X}_{d_2}$

Если $d_3 < 4$, то d_3 в рим.с.с. = $\underbrace{I \dots I}_{d_3}$

Если $d_1=4$, то d_1 в рим.с.с. = ID

Если $d_2=4$, то d_2 в рим.с.с. = IL

Если $d_3=4$, то d_3 в рим.с.с. = IV

Если $(d_4 > 4)$ и $(d_4 < 9)$, то d_1 в рим.с.с. = $\underbrace{D I C}_{d_1=5}$

Если $(d_2 > 4)$ и $(d_2 < 9)$, то d_2 в рим.с.с. = $\underbrace{X \dots X}_{d_2=5}$

Если $(d_3 > 4)$ и $(d_3 < 9)$, то d_3 в рим.с.с. = $\underbrace{V I \dots I}_{d_3=5}$

Если $d_1=9$, то d_1 в рим.с.с. = CM

Если $d_2=9$, то d_2 в рим.с.с. = XC

Если $d_3=9$, то d_3 в рим.с.с. = IX

$x_2 = \overline{d_1 d_2 d_3}$ (x_2 состоит из последовательных чисел d_1, d_2 и d_3 в римской с.с.)

Получается, что наше число m в римской системе счисления равно $\overline{x_1 x_2}$ (~~x_1~~ x_1 и x_2 записываются последовательно).



Таким образом, мы перевели восьмеричное число n в десятичную систему счисления.

Задача

26	27+	28+	29+	30+	31
+					32+
25	10	11+	12+	13	32+
24+	9+	2	3	14	33+
23+	8+	1	4+	15	34+
22+	7	6+	5	16+	35+
21	20+	19+	18+	17	36+

решение

Плюсиком отметим в таблице те квадраты, для построения которых необходимо 2 рейки.

Мы видим, что все квадраты, кроме тех, которые имеют вид n — темные шема,

кроме тех, которые имеют вид n (номер квадрата) = $i \cdot i + 1$, строятся с помощью двух реек.

Это происходит от того, что все квадраты

вида $n = i \cdot i + 1$ ~~строится~~ присоединяются к квадрату $i \cdot i$, но есть образуют угол. Все квадраты, образующие угол строятся с помощью 3 реек. А все квадраты имеющие темные номера, но не имеющие вида $n = i \cdot i + 1$ строятся уже к углу и дополняют его до нового угла или до прямоугольника, поэтому для их построения необходимо 2 рейки.

Все квадраты с темными номерами и вида $n = i \cdot (i-1) + 1$ ~~распределены~~ строятся к прямоугольнику $i \times (i-1)$ и ~~распределены~~ его до угла, т.е. для них необходимо 3 рейки. Все остальные темные квадраты ~~распределены~~ ^{строится} до угла, поэтому для них нужно 2 рейки. (Камешки из правых 1-ый квадрат, и темн — 4 рейки).

Напишем программу на языке Паскаль:

```

program z1;
var n, i, s: longint;
begin
  readln (n);
  if n mod 2 = 0 then
    begin
      s := 0; { s-считчик }
    end
  end

```





```

for i:=1 to n do
if (i+1=n) then s:=s+1;
if s=0 then writeln (2) else writeln (3);
end;
end (n<>1)
if (n mod 2=1) then begin
s:=0;
for i:=1 to n do
if (i*(i-1)+1=0) then s:=s+1;
if s=0 then writeln (2) else writeln (3);
end; if n=1 then writeln (4);
end.

```

Задача 3.

Решение.

Решим эту задачу на языке программирования Pascal.

```

program z1;
var a, n, b, i: longint;
function prost(x, y: longint): longint;
var j, k: longint;
begin
k:=0;
for j:=1 to x+y do
if (x mod j=0) and (y mod j=0) then k:=k+1;
if k=1 then prost:=1 else prost:=0;
end;
begin
readln (n, a);
b:=0; b:=-1;
for i:=0 to n do
if (a*i) mod n=1 and (prost(a, i)=1) then b:=i;
if b=-1 then writeln ('не существует') else writeln
(b);
end.

```

{ в функции: если один элемент 1, то
ища взаимнопростые, если нет, то
не взаимнопростые }

a - данное число
b - число, которое мы находим
n - модуль





гот. лист №2

Задача 5.

Решение.

Решим эту задачу на языке Pascal. Нам даны 3 одномерных массива a, b, c и n -число экспериментов. С помощью сортировки посплошным все данные в массиве c по убыванию и соответственным образом изменить данные в массивах a и b .

```

program z1;
var a, c: array [1..100000] of longint; b: array [1..100000] of string;
n, i, j, k, k1, k2: longint;
begin
  readln(n);
  for i:=1 to n do
    readln (a[i], b[i], c[i]);
  for i:=1 to n-1 do
    for j:=i+1 to n do
      begin
        if c[j] > c[i] then begin
          k:=c[i];
          c[i]:=c[j];
          c[j]:=k;
          k1:=b[i];
          b[i]:=b[j];
          b[j]:=k1;
          k2:=c[i];
          c[i]:=c[j];
          c[j]:=k2;
        end;
      end;
  for i:=1 to n do
    write (a[i], b[i], c[i]);
  writeln
end.

```



Задача 2.

Решение

Любую цифру рационального действительного числа можно написать в двоично-десятичной с.с. (четырёхбитовый код двоичной кода достаточен для чисел от 0_2 до 1111_2 , т.е. от 0 до 15). Значит любое рациональное число можно записать в двоично-десятичной с.с., кроме отрицательных чисел, т.к. не хватит разрядов для "-".

~~А вот в двоично-десятичной с.с. кроме отрицательных чисел...~~

Олимпиада школьников «Надежда энергетики»

№ группы

№ группы

Вариант № 7103

0А 30-20

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Исрапилов

ИМЯ Махач

ОТЧЕСТВО Исрапилович

Дата рождения 05.01.1999г.

Класс: 10

Предмет Информатика

Этап: Заключительный

Работа выполнена на 2 листах

Дата выполнения работы: 15.03.2015г.
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№1



Найдем такие два последовательно идущие в ряду Фибоначчи числа x_n и x_{n-1} , $x_{n-1} < x_n$, для которых выполняется условие $x_{n-1} \leq x < x_n$, где x - число, которое мы хотим представить в Фибоначчиевой системе счисления. Из x вычтем x_{n-1} и запишем это x_{n-1} . Если после вычитания $x - x_{n-1} = x_1$ останется число $x_1 > 0$, то мы повторим вычитание действие, приняв $x = x_1$, до тех пор, пока очередное x_i не станет равно 0. Запишем в ряд все числа Фибоначчи, до первого x_{n-1} . Запишем, соответствующими x_{n-1} , записав единицу, на оставшихся нули. Представив полученную последовательность нулей и единиц в отрезке поперек, получим желаемое представление заданного числа в Фибоначчиевой системе счисления.

№2



Запишем предложенное на проверку число как p . Для уравнения $p-1 = 2^s \cdot d$ найдем s и d . Будем делить $p-1$ на 2, пока не получим нечетное число. Оно и будет d , а количество делений - s . Выпишем $t = \log_2 p$ и отбросим от t дробную часть. Теперь t раз случайно выберем a из $\{2, \dots, p-2\}$. Проверим для каждого a справедливы ли две условия:



1) $a^d = 1 \pmod{p}$

2) существует τ , $0 < \tau \leq s-1$, для которого $a^{t-d} = -1 \pmod{p}$

где $t = 2^{\tau}$.
 Для проверки первого условия вычислим a^d и проверим равенство этому числу $1 \pmod{p}$. Для проверки второго вычислим все значения a^{t-d} при $\tau \in [0, s-1]$, и для каждого значения проверим равенство $a^{t-d} = -1 \pmod{p}$.
 Если для всех a хотя бы одно из условий справедливо, то, вероятно, число простое, в противном случае оно явно составное.

№3

Любое число представляется в виде суммы зрелых чисел. Если представить M в виде суммы некоторого ряда чисел m_i , то $M = m_1 + m_2 + \dots + m_n = M^m \cdot M^{m_2} \cdot \dots \cdot M^{m_n}$.
 При m_i достаточно малых ($m_i \leq 3$), выражение представляется в виде простого произведения. Теперь необходимо представить M в виде ряда чисел m_i . Для этого запишем M в виде суммы чисел в последовательности по 2 числа в каждой. Тогда M можно представить в виде суммы 3 чисел - 2 одинаковых $M/3$ и одного $M/3$, который мы запишем 3 раз. Тогда $M = 3 \cdot (M/3) = 3 \cdot M/3$.

№2 - неБ
№3 - неБ



показательности, умножив на 2 столько раз, сколько мы ранее поехали. По-второму вычитаемые операции до тех пор, пока все m в m не станут достижимо малы (< 3). После этого, пройдя все операции в уравнении $N^m = N^{m_1} \cdot N^{m_2} \dots N^{m_k}$, найдем N^m

Возьмем первое число на листе 1 за m , и будем последовательно сравнивать все числа на листе с m . Если 1-е число $< m$, то мы заменим m на меньшее из этих чисел на 1-е, совершив 1. Таким образом пройдя лист 1, заменим m на 2-е число m первым из оставшихся m . Вернемся к 1-му числу с листе 1. Повторим, пока на листе 1 нет каждого из чисел. После завершения, на листе 2 будут упорядоченные в возрастающем порядке все числа листа 1.

Список - упорядоченный набор элементов, в котором адрес следующего (иногда и предыдущего) элемента хранится в самом элементе. Уникальная граница списка не существует. Достоинство списков в возможности хранить неограниченное количество элементов, т.к. сам список, по сути, хранит лишь адрес 1-го элемента. Недостаток в том, что для выбора i -ого элемента списка нужно пройти весь список от 1-ого и до i -ого элемента. Такая структура идеально подходит для хранения данных в этой задаче.

Для решения задачи поехали участок $1 \cdot K$ на $\frac{1 \cdot 1000}{K}$ разов по K км. Каждый раз поехали на $\frac{K \cdot 1000}{\sqrt{3}}$ км. Каждая ячейка - элемент списка, представляющего ряд. Каждый раз $\frac{1}{\sqrt{3}}$ элемент списка о весь участке. Если разов, таким образом - элемент обьема участка. В каждой ячейке максимум одна ель. В элемент списка, соответствующий ячейкам, будем заносить число, соответствующее количеству елей в этой ячейке, если растущей. Если елей там нет, заносим 0. Последовательно занесем необходимые данные в список. Сформируем еще один список, с количеством элементов $N-N+1, N-N+1$ раз пройдем по первому списку (общему, среднему), по разу для каждого элемента списка 2. Каждый проход будем подсчитывать количество элементов в 1 списке, которое хранит числа, равные номеру элемента в списке 2 + $N-1$, и заносить полученное количество в элемент списка 2. Таким образом, в результате список 2 будет хранить количество елей в каждой ячейке от N до N в настоящее время элемент раз номерами от 1 до $N-N+1$ соответственно.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 3093

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ КОВАЛЕВ

ИМЯ Даниил

ОТЧЕСТВО ЮРЬЕВИЧ

Дата рождения 25.04.2001

Класс: 9

Предмет Информатика

Этап: Заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады:

Д.Ковалев

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



79

В первом решетке такая четверка есть: 2, 3, 5, 7. В других же решетках простыми могут быть лишь простые числа, т.е. делимые на 1, 3, 5, 7, 9, но число, делимое на 5, делится на 5, и оно составное (также оно может быть равно 5, но это рассмотренный случай с первой решеткой). Таким образом, будем проверять у каждой решетки на простоту только 4 числа из 10. Реализация алгоритма на C++:

```
#include <iostream>
#include <math.h>

using namespace std;

bool Prost (int a)
{
    for (int i =
    if (a==1)
        return false;
    else if (a==2 || a==3 || a==5 || a==7 || a==11 || a==13 || a==17)
        return true;
    if (a%2==0 || a%3==0)
        return false;
    for (int i=5; i<=sqrt((double)a); i++)
        if (a%i==0)
            return false;
    return true;
}

int main()
{
    cout<<" 2 3 5 7"<<endl;
    for (int i=1; i<50000; i++)
        if (Prost (10*i+1) && Prost (10*i+3) && Prost (10*i+7) && Prost (10*i+9))
            cout<< 10*i+1<<" " << 10*i+3<<" " << 10*i+7<<" " << 10*i+9<<endl;
    return 0;
}
```

Для ускорения проверки на простоту небольшие простые числа проверили вручную, также сразу проверили делимость на 2 и на 3. Далее стандартно будем перебирать все числа до \sqrt{a} , если какое-то из них - делитель a , то оно составное. Число - простое. Это реализуется в функции Prost();
Нахождение четверки и вывод на экран - в функции main().

+



№5

$$\begin{array}{r}
 \cdot 1 \cdot \\
 \times 3 \cdot 2 \\
 \hline
 \cdot 3 \cdot \\
 + 3 \cdot 2 \cdot \\
 \cdot 2 \cdot 5 \\
 \hline
 1 \cdot 8 \cdot 3 \cdot
 \end{array}$$

Понятно, что во втором множителе последняя цифра 0 (пусть это x , имеем $3+x=3$ или $4+x=3$ (если был переход через разряд), откуда $x=0$ или $x=9$ (в случае перехода через разряд), но переходе быть не могло, следовательно, $x=0$). Первая цифра ~~второго~~ ^{первого} множителя — это 1 (в сумме только этой цифрой получается 1).

При умножении $3 \cdot 1^* = 12^*5$, откуда последняя цифра первого множителя — это 5, перебором получаем, что первая цифра — это 4, тогда произведение 1245. Имеем:

$$\begin{array}{r}
 \times 415 \\
 \times 3 \cdot 2 \\
 \hline
 830 \\
 3 \cdot 20 \\
 1245 \\
 \hline
 1 \cdot 8530
 \end{array}$$

Из-за перехода через разряд имеем вторую цифру второго множителя: $x+4+1=8$, $x=3$, откуда вторая цифра второго множителя $3320:415=8$, и весь ряд будет выглядеть следующим образом:

$$\begin{array}{r}
 415 \\
 \times 382 \\
 \hline
 830 \\
 3320 \\
 1245 \\
 \hline
 158530
 \end{array}$$



№6

Переводим буквы по стандартному алгоритму, умножив цифру на значение соответствующего разряда, например, $132_4 = 1 \cdot 4^2 + 3 \cdot 4^1 + 2 \cdot 4^0 = 16 + 12 + 2 = 30$. Предварительно буквы будут переводить в числовой формат (выраим это символом строки), а стандартные цифры переведем в соответствующие числа (A в 10, B в 11, C в 12, ..., F в 15). Полученное в конце число приведет к следующему шагу. Реализация не с++:



```
#include <iostream>
#include <string>
using namespace std;
int toNumber(char c)
{
    if (c=='0')
        return 0;
    else if (c=='1')
        return 1;
    else if (c=='2')
        return 2;
    else if (c=='3')
        return 3;
    /* здесь также не
    обрации с цифрами от
    4 до 9 */
    else if (c=='F')
        return 15;
    else
        return -1; // ошибка при вводе числа
}
```

```
int power(int a, int b) // возвести в степень
```

```
{
    int s=1;
    for (int i=0; i<b; i++)
        s*=a;
    return s;
}
```

```
int main()
```

```
{
    string kx;
    int q;
    cout << "Введите kx: ";
    cin >> kx;
    cout << "Введите q: ";
    cin >> q;
    int ans=0;
```

```
for (int i=0; i<kx.length(); i++)
    ans += toNumber(kx[i]) * power(q, kx.length()-1-i); // будем считать, что ошибка при
// вводе числа нет, иначе просто
// добавим оператор if (toNumber(
// kx[i]) != -1)
string s = (string)ans;
cout << s;
return 0;
}
```



N3

```
/* Ввод исходных данных */
int d=p-1, s=0;
while (d%2==0)
{
    d/=2;
    s++;
}

```



Каждым d и s . Для этого будем делить $p-1$ на 2, пока оно ещё делится, и при делении, увеличивать s на 1. Код на C++ для этого участка представлен ниже. При проверке условия $a^d = 1$ (тогда p будет кратно p при умножении, не а браз остаток от деления на p , этот остаток нечетно-инский, и подумывающая величина делится будет быть равна 1.

Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7113

ФАМИЛИЯ КОРСКОВ

ИМЯ АЛЕКСЕЙ

ОТЧЕСТВО ВЛАДИМИРОВИЧ

Дата рождения 30.01.98

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

Дата выполнения работы: 15.03.15
(число, месяц, год)

Подпись участника олимпиады: Корс

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



Задача №1.

Опишем наш алгоритм на языке pascal:

Var x, pp, p, n, r, ch: integer; {x - данное число, pp, p, n - числа Фибоначчи, r - для сдвига разрядов, ch - результат}

begin
read(x); {читаем число, которое надо перевести}
if (x < 0) or (x >= abs(x)) then {проверяем чтобы число было целым и неотрицательным}
writeLn('некорректные данные');

else

begin

pp := 1;

p := 1;

n := 2;

while (n <= x) do

begin

pp := p;

p := n;

n := p + pp;

end;

n := p;

p := pp;

pp := n - p;

r := 1;

while (n >= 1) do

begin

if (x <= n) then

begin

ch := ch * r + 1;

r := r * 10;

x := x - n;

end;

if (x > n) then

begin

ch := ch * r;

r := r * 10;

end;

n := p;

p := pp;

pp := n - p;

end;

ch := ch div 10;

writeLn(ch);

end;

end.

{начало последовательности Фибоначчи}
{для нашего алгоритма нам нужно знать эти последовательные числа Фибоначчи}

{этими циклом мы найдем максимальный разряд, в который помещается наше число. Этот разряд хранится в переменной r}

{делаем так, чтобы макс. разряд хранился в n}

{переменная r нужна для сист. результата}

{этот цикл для поиска результата}

{если число входит в разряд, умножаем на r, чтобы сдвинуть предыдущие разряды влево, прибавляем 1 и вычитаем из числа значение разряда}

{если число не входит в разряд, то просто сдвигаем разряды и умнож. r}

{находим последний подходящий разряд в нашей сист. чисел}

{в результате выведем число, которое будет иметь нужный разряд (где 1) и тогда это исправит деление на 10}

{вывод результата}

Внешний вид программы



Задача №5.

```

Var a: array [1..1000] of integer; { массив для чисел }
m, n, i, j: integer; { m, n - размер массива, i, j - переменные для циклов }
Begin
  read(m, n); { считываем размер }
  if (m <= 0) or (n <= 0) then { проверяем данные на правильность }
    writeln('данные некорректные')
  else
    begin
      for i := 1 to m do
        for j := 1 to n do
          read(a);
      for i := 1 to m * n do { считываем данные в одно-
        read(a[i]); { мный массив }
        for j := 1 to m * n do
          begin
            for j := 1 to m * n do
              begin
                if (a[j] < a[j+1]) then { проверяем правильно
                  begin { ли стоит число }
                    d := a[j]; { если неправильно,
                    a[j] := a[j+1]; { то меняем их }
                    a[j+1] := d;
                  end;
                end;
              end;
            end;
          end;
        for i := 1 to m * n do { вывод результата }
          write(a[i], ' ');
        end;
      end.

```

Задача №3.

```

Var b: array [1..1000] of boolean; { массив для результатов проверки }
p, d, s, t, i, pr, pr2, p3, a, j, l: integer; flag: boolean;
Begin
  read(p); { считываем число }
  if (p <= 2) then
    writeln('данные некорректные') { проверяем данные на правильность }
  else
    begin
      d := p - 1;
      while s := 0;
      while (d mod 2 = 0 d div 2 = d / 2) do { проверяем делится ли число на 2 и находим d и s }
        begin
          d := d div 2;
          s := s + 1;
        end;
      продолжение на след. листе...
    end;

```




Задача №3

```

t := log2(p!);
for i := 1 to t do {в этом цикле проверяем t чисел a}
begin pr := 1; pr2 := 1; pr3 := 1; flag := false;
  a := randomize(p-2)+1; {налогично числу a из задачи}
  for j := 1 to d do {в этом цикле наложим a в степени d}
    pr := pr * a;
  for j := 0 to s-1 do {в этом цикле перебираем все r}
    begin
      for l := 1 to j do {в этом цикле наложим k}
        pr2 := pr2 * 2;
      for l := 1 to pr2 * j do {в этом цикле наложим
        pr3 := pr3 * a; {a в степени k-d}
        if (pr mod p = 1) or (pr3 mod p = p-1) then
          flag := true; {проверяем выполняются
            ли условия}
        end;
      a[i] := flag;
    end;
  flag := true;
  for i := 1 to t do
    if (a[i] = false) then
      flag := false;
    if (flag = true) then
      writeLn('вероятно простое') {вывод результата}
    else
      writeLn('составное');
    end;
  end;
end;

```

Записываем результат проверки в массив для каждого a
 Если где-то хотя бы 1 a проверка не пройдена, то мы делаем вывод, что число составное
 и выводим результат

Задача №4.

```

Var a, b: array[1..100] of integer; {a - массив для цифр N, a -
m, n, i, kn, p, r, j, l, ps: integer; {b - массив для цифр 'отседа'}
{p - переменная в след. разряде, ps - переменная в
текущем разряде}
Begin
  read(m, n); {считываем цифры}
  While (n > 0) do {записываем цифры числа n
    begin {в массив}
      i := i + 1;
      b[i] := n mod 10;
      n := n div 10;
    end;
    kn := i; {kn - кол-во цифр в числе n}
    p := 0; ps := 0; {p - переменная в след. разряде}
  продолжение на след. листе...

```



Задача 5.

$r := kn$; $\{r\}$ - кол-во цифр в ~~ответе~~ промежуточном знач. }
 for $i := 1$ to m do $\{ \text{мы умножаем } m \text{ раз} \}$
 begin

for $j := 1$ to kn do $\{ \text{умножаем на каждую цифру числа } n \}$
 begin

~~$p := 0$~~ r
 for $l := 1$ to kn do $\{ \text{в этом цикле мы умножаем } a[l] \}$
 begin $\{ \text{с каждой цифрой промежуточ. знач.} \}$

$p := a[j] \cdot a[l] \text{ div } 10$

$a[l] := a[l] + a[j] \cdot b[j] \text{ mod } 10 + p$

if $(a[l] \geq 10)$ then

$p := p + 1$

end; $ps := p$; end;

if $(ps > 0)$ then

begin

while $(ps > 0)$ do

begin

~~$k := k + 1$~~ $r := r + 1$

$a[r] := ps \text{ mod } 10$;

$ps := ps \text{ div } 10$;

end;

end;

end;

for $i := r$ down to 1 do
 write $(a[i])$;

end.

$\{ \text{в результате мы получили от-} \}$
 $\{ \text{вет, но он будет перевернут,} \}$
 $\{ \text{выводим массив } a \text{ в обр. порядке} \}$

$\{ \text{если } p > 0, \text{ то нам} \}$
 $\{ \text{нужно расширить на-} \}$
 $\{ \text{ше число и дописать} \}$
 $\{ \text{цифру знач. } p \text{ делен на} \}$
 $\{ \text{разряды, и занеси-} \}$
 $\{ \text{ваем } p \text{ на знач.} \}$
 $\{ \text{в наш массив} \}$

Задача 6.

Информацию о плане рассадки мы будем хранить в массиве. У нас их будет два. Первый (внутренний) будет хранить информацию о полях в L км. второй (внешний) будет хранить информацию о каждом внутреннем участке в K км. Мы будем проходить по этим участкам и там найдем поля, но будем анализировать кол-во ярусов. Заведем также массив с индексацией от M до N . И, проанализировав кол-во ярусов у внутренней вли, прибавим $+1$ к тому элементу массива, индекс которого равен кол-ву ярусов у этой вли. В результате мы получим массив, содержащий информацию о полях с каждым числом ярусом. После считывания информации проверим ее на корректность. M, N, K и L должны быть больше 0.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 1102

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Космин

ИМЯ Михаил

ОТЧЕСТВО Юрьевич

Дата рождения 14.06.1999

Класс: 10

Предмет Информатика

Этап: Заключительный

Работа выполнена на 3 листах

Дата выполнения работы: 24.02.15
(число, месяц, год)

Подпись участника олимпиады: Космин

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№5

Присвоить некоторой переменной I значение результата, стоящего на первой строке таблицы, сравнить его с каждой следующей, и, если сравниваемый результат ^{сравниваемого} меньше значения переменной, то присвоить переменной значение результата; Переменной II присвоить значение один. Так, пройдя до конца таблицы, переписать строку, в которой стоит результат, присвоенный переменной, на первом месте, и переписать всю таблицу вту. Присвоить переменной II значение переменной II плюс единица.

При последующих присваиваниях переменной I, первый присваиваемый результат брать из строки с номером n, где n — значение переменной II. Номер строки — её номер, при подсчете идет строки сверху вниз. Закончить алгоритм при достижении переменной n значения номера последней строки, при подсчете сверху.

№3

```

program z3;
var a, n, b, i, k : longint; label 1;
begin
  readln(a, n); b := 1;
  k := 1;
  for i := 1 to n do
    if ((a mod i) = 0) and ((n mod i) = 0) then k := k + 1;
  if k > 0 then begin writeln("k существует"); goto 1; end
  else begin
    for i := 1 to a do begin
      if ((a - i) mod n) and ((a * i) = 1) then b := i;
    end; end;
if b = 1 then if b = 1 then writeln("не существует") else writeln(b);
1:
  readln;
end.

```



№2

Приманение двоично-десятичной системы счисления целесобразно, если:

- 1) Требуется узнать кол-во цифр в записи числа десятичного вида, не переводя его самому в десятичную систему счисления.
- 2) Если требуется вывести какой-либо разряд числа, то это можно сделать не переводя всего числа в десятичную систему счисления, переводя лишь четвертый битный двоичный код нужного разряда.
- 3) Более легкий перевод из десятичной в двоичную, так и из двоично-десятичной в десятичную, так и обратно, нежели в двоичную в шестнадцатеричную.

№1

```

программ 21;
var
begin
  readln (a); k:=0; D:=1;
  while a <> 0 do begin
    b:=a mod 10;
    a:=a div 10; k:=k+1;
    D:=b;
    for i:=1 to k do
      D:=D*8
    S:=S+D
  end;

```

⊖

⇒ Перевести ~~в десятичную~~ число из восьмеричной с.с. в десятичную.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

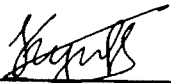
ФАМИЛИЯ КУПРИАНОВ
ИМЯ КИРИЛЛ
ОТЧЕСТВО ИГОРЕВИЧ

Дата рожу 24.05.1998 Класс: 11

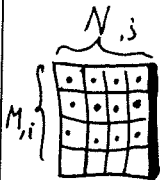
Предм. ИНФОРМАТИКА Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 2 листах Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ОБЩИЙ ВИД РЕШЕНИЯ: нули макс. 71-т, боимали нули 1 пока не дойдём до след. нуля макс.



№5.

- 1) Все числа с блокнотного листа записать в матрицу $X (M \times N)$, где левый верхний 71-т блок листа будет в ячейке $(1, 1)$ матрицы.
- 2) Найти максимальный 71-т P и минимальный 71-т Q . Поменять местами 71-т P и Q , т.е. поменять $(i, 1)$ поменять местами 71-т P и Q , т.е. поменять $(i, 1)$. Поменять Q и 71-т $(M; N)$. Записать в переменную K роль. $K := 0$.
- 3) Начиная со второго столбца, а строки начинаем проверять элемент $X(i, j)$ равен $P - (K+1)$. Если такое число найдено, то по нулю поменять местами с числом, следующим после P . После нахождения адекватно малому числу, поменять его местами, столбцам на 2 позиции дальше P . Следующее число P поменять с тем, что на 3 позиции дальше P .
- 4) Продолжить до тех пор, пока сравнение не дойдёт до числа Q . Число упорядочено.

№6.

Общее кол-во видов = $N - M + 1$. В массив $A [1..(N - M + 1)]$, где $A [1]$ - кол-во видов этой вида M . $A [2]$ - кол-во видов этой $M + 1$; и т.д. $(N - M + 1)$ - кол-во видов этой вида N .

Пусть поле $K \times L$ представляет собой матрицу $(K \times L)$ с некоторыми элементами, значение которых = 0, т.е. по условию, не все они есть единицы. Начнём заполнять массив A . $A [1] := B(1, 1)$, массив A должен быть заполнен полностью ролями числами, значения матрицы B . Если попадет значение 0 то его писать в массив A не нужно. Затем, когда массив A полностью заполнен, берём $A [1]$ и считаем сколько разок в матрице B равно элементу $A [1]$. Сколько наступило, столько и пишем в $A [1]$.

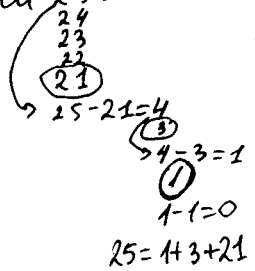
Теперь в $A [1]$ содержится не вид M , а кол-во этой вида M . Аналогично, производим ту же процедуру с $A [2], A [3], \dots, A [N - M + 1]$.
 затем введем: "Если вида M есть $A [1]$ "
 "Если вида $(M + 1)$ есть $A [2]$ "
 "Если вида N есть $A [N - M + 1]$. Посчитано.

5	6	10	1	0	7
10	1	7	5	4	8
1	10	1	5	8	8
7	10	1	6	10	6

№7.

- 1) Проверить, является ли введенное число членом Фибоначчи. Если да, то вывести 1 и столько нулей, сколько чисел Φ стоит до него. Если нет, то вывести 0.
- 2) Вывести из N количество Φ и продолжить до получения числа Φ .
- 3) И так далее пока не получится 0. (Т.е. числа Φ , которое мы боимали, следует записать в массив, например, на примере эти числа 21; 3 и 1.
- 4) На полученных членах числа записать сумму, если есть нули, числа Φ 1 и 3 и 21. 1 стоит на 1 месте, 3 стоит на 3 месте, 21 стоит на 7 месте. Соответственно, по 1, 3 и 7 летам ставим единицы в исходном числе, а на 2, 4, 5, 6 - нули.

ПРИМЕР: ввели 25.





Очевидно, результат будет представлять из себя орошиа числа
 Тоше числа удобнее хранить в массиве. Нужно уметь считать M как
 наибольшее число (маленькое) в отечени $1, 2, 3, \dots, n$, чтобы
 удобнее было считать. Например, $3^8 = 3^2 \cdot 3^2 \cdot 3^2 \cdot 3^2 = 3^{2+2+2+2} = 3^2 \cdot 3^2 \cdot 3^2 \cdot 3^2$.
 Считая маленькими корнями, размещали результаты в массиве.
 А. первую цифру в первую ячейку, вторую - во вторую и т.д.



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7092

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Лысков

ИМЯ Леонид

ОТЧЕСТВО Сергеевич

Дата рождения 18.10.1999

Класс: 9

Предмет Информатика

Этап: Заключительный

Работа выполнена на 3 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады: Леонид

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№1

$$1 \cdot 1! + 2 \cdot 2! + \dots + n \cdot n! + 1 = (n+1)!$$

при вычитании данных условия
можно ~~что-то~~ было можно
выразить через эту сумму с.
и только 1 способ

$$(n+1)! = n! \cdot (n+1)$$

$$n! \cdot (n+1) - n \cdot n! = n! \Rightarrow$$

при $n \geq 1$

$$\Rightarrow 1 \cdot 1! + 2 \cdot 2! + \dots + n \cdot n! + 1 = (n+1)!$$

с.с. является рекурсивной.

Ответ: с.с. f является рекурсивной.

№2

τ - GMT

p - раз

k - коэффициент при переборе часов, может быть положительным, отрицательным и 0.

τ

p

c	c	c	c	c	c	c
k	k	k	k	k	k	k

$$t_{pc} = \tau + p + k$$

№3

```

var p, i, a, b, k: longint;
begin
  readln(p); k := k 0;
  for i:=1 to (p-1) do begin
    a:=i; b:=p;
    while a <> b do begin
      if a > b then a:=a-b;
      if b > a then b:=b-a;
    end;
    if a=1 then k:=k+1;
  end;
  writeln(k);
end.

```

алгоритм - Pascal



N5

$$10000 : 77 = 129,8... \Rightarrow \min = 130 \cdot 77 = 10010$$

$$100000000 : 77 = 1298701,29 \Rightarrow \max = 1298701 \cdot 77 = 99999977$$

var i

begin

~~write~~i := 10010; ~~writeln(i);~~

while i < 100000000 do begin

i := i + 77;

writeln(i);

end;

end.

N6

$1 + \epsilon_M = 1$ где компьютерное число $< \epsilon_M, = 0$.

var e, c: real;

begin

e := 1; c := 1;

while e > 0 do begin

c := e;

e := e/2;

end;

writeln(c);

end.

N7

var a, b: array[-20..220] of longint; k, c, d: longint;

begin

a[-20] := 1; d := 1;

for c := 1 to 44 do begin

for k := 1 to d do begin

~~a[k] := a[k] * c~~

a[k+1] := a[k+1] + a[k] div 10;

a[k] := a[k] mod 10;

if a[k+1] > 0 then d := d + 1; end;



```

end; k:=x
while a[k] > 0 do begin
  k:=k+1;
  k:=d downto 1 do begin
    if a[k]=0 then k:=d;
  while a[k]=0 do begin
    d:=d-1; k:=d; p:=0
  for p:=k downto 1 do begin
    if a[p]=4 then
      p:=p+1;
    end;
  writeln(p);
  end.

```

(F)

N4

$N = \overline{a_1 b_1 c_1 \dots x_1 y_1 z_1} \quad k := 1$
 $M = \overline{a_2 b_2 c_2 \dots x_2 y_2 z_2}$

$$N = a_1 \cdot 10^x + b_1 \cdot 10^{x-1} + \dots + y_1 \cdot 10 + z_1$$

$$M = a_2 \cdot 10^x + b_2 \cdot 10^{x-1} + \dots + y_2 \cdot 10 + z_2$$

~~$$z_1 \cdot z_2 + z_1 \cdot y_2 \cdot 10 + \dots + a_1 \cdot a_2 \cdot 10^x$$~~

~~$$z_x = z_1 \cdot z_1$$~~

~~$$N[1..1000] \quad c[x] = c[x]$$~~

~~$$M[1..1000]$$~~

~~$$c[x+y] = k[x] \cdot d[y];$$~~

~~$$c[x+y+1] := c[x+y] \text{ div } 10;$$~~

~~$$c[x+y] := c[x+y] \text{ mod } 10;$$~~

```

k:=1;
for i:=1 to M do
  k:=k*N { отсюда }
writeln(k);
end.

```

(T)

$c[1000..1]$
 M раз

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ МАКСУДОВ

ИМЯ АНАТОЛИЙ

ОТЧЕСТВО ВЛАДИМИРОВИЧ

Дата рождения 30.10.1996

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



1. var
 x, c, a1, a2: Int64; {на случай больших чисел}
 begin
 read(x);
 if x < 0 then begin write('-'); x := -x; end;
 {x := -x можно заменить на x := abs(x)}
 a1 := 0; a2 := 1;
 while a1 + a2 ≤ x do begin
 c := a2; a2 := a1 + a2; a1 := c; end;
 {можно обойтись без вспомогательной переменной c:}
 {a2 := a1 + a2; a1 := a2 - a1;}
 if a1 = 0 then write(0) else begin
 repeat
 if x ≥ a2 then begin
 write(1); x := x - a2; end else write(0);
 c := a1; a1 := a2 - a1; a2 := c;
 {также можно было обойтись без вспомогательной переменной}
 {c: a1 := a2 - a1; a2 := a2 - a1;}
 until a1 = 0
 end; end.
 {Язык программирования: Pascal}

3. var
 j, i, p, s, d, c, a, t: Int64; e: boolean;
 begin
 read(p); if p ≤ 2 then write('ошибка ввода');
 if p mod 2 = 0 then write('составное') else begin
 d := p - 1; s := 0; e := false
 while d mod 2 = 0 do begin
 inc(s); d := d div 2; end;
 t := 1; c := 2; while ~~p~~ p > c do begin
 c := c * 2; inc(t); end; dec(t);
 {считаю, что t ≤ log₂ p}
 {t можно было присвоить t := trunc(ln(p) / ln(2))}
 For i := 1 to t do begin a := random(p - 1);
 {a ∈ [0; p-1]} c := a
 For j := 2 to d do c := (c * a) mod p;
 if not ~~(c mod p = 1) or (c mod p = p - 1)~~ then {перенос случайя}
 write('вероятно составное') {r = 0}



```

else begin
  for r:=1 to s-1 do e:=(c*c) mod p; begin c:=(c*c) mod p;
  if c mod p = p-1 then break; end;
  if c mod p = p-1 then write('вероятно простое');
  else write('вероятно составное');
  if e then write('составное')
  else write('вероятно простое'); end; end.

```

5. После чтения матрицы (для которой ранее выделена память), проходя по строкам матрицы и заносим числа в массив ~~массив~~ из строк. Каждая строка - это число, составленное из расположенных цифр в матрице; в пустые клетки матрицы заносим ~~символ~~, в клетки с $2-1$ - число -1 (число -1 будет заноситься в строку как символ $'-1'$, а число 10 будет считаться пустой клеткой при прохождении матрицы). Инициализация следующая:

```
var b: array[1..M, 1..N] of byte;
```

```
a: array[1..(M+1) div 2 * N] of string;
```

После прохода по всей матрице сортируем массив строк a с помощью следующей процедуры:

```

procedure qsort (a:mas, first, last: longint);
var l, r: longint;
    k: string;
begin
  f:=first; l:=last; k:=a[(f+l) div 2];
  repeat
    while a[f] > k do inc(f)
    while a[l] < k do inc(l)
    if l >= f then begin k:=a[f];
      a[f]:=a[l]; a[l]:=k; inc(f); dec(l); end;
  until f >= l
  if l > first then qsort(a, first, l);
  if f < last then qsort(a, f, last);
end;

```

Данные выводи массив a и, если возможно, выводи несколько чисел в одну строку (общая длина



чисел ~~и~~ + именованных между ними не превышает длины не превышает M ~~и~~ ~~и~~)

6. На 1 км^2 помещается $5 \cdot 10^5$ семян.

Храним данные будем следующим образом:

$a: \text{array}[1..K, 1..L] \text{ of } \text{set}$;

где описывается тип «set»:

type

set = set of

~~$b: \text{array}[M..N] \text{ of } \text{longint}; \text{end};$~~

~~алгоритм: от N до~~

~~for $i := 1$ to K do~~

~~for $j := 1$ to L do for $e := 1$ to N do~~

~~$s[e] := a[i, j].b[e] + s[e];$~~

(в массиве b хранится количество семян каждого вида на ~~определенной~~ конкретной площадке в 1 км^2)

(s - массив сумм, т.е. общее количество семян каждого вида, которые после окончания алгоритма необходимо вывести)

(e, i, j - переменные такого же типа, что и M, N, K, L)

4. Если бы числа были не столь большими, то достаточно было бы

if $N=0$ then write(0) else

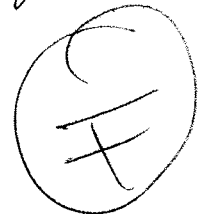
if $N=1$ then write(1) else

if $M=0$ then write(1) else

~~далее цикл от 1 до M $M \cdot C := N$~~

for $i := 1$ to M do $C := N \cdot C$

(линейная арифметика - умножение)



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7102

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ МИХАЙЛОВ

ИМЯ НИКИТА

ОТЧЕСТВО МАРАТОВИЧ

Дата рождения 27.08.1998

Класс: 10


Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№4

```

var a, n, i, s: integer;
begin
  readln(n); a := 1; s := 0;
  while a * a <= n do inc(a);
  i := a - 1;
  while i >= 1 do begin
    s := s + 4 * i;
    i := i - 2;
  end;
  while i >= a - 2;
  while i >= 1 do begin
    s := s + 4 * i;
    i := i - 2;
  end;
  for i := 1 to n - (a - 1) * (a - 1) do begin
    if (i = 1) or (i = a) then s := s + 3;
    else s := s + 2;
  end;
  writeln(s);
end.

```



№5

```

var n, i, kod, k, a, b h: integer; a, s, c: string;
Mass: array[1..100] of Record

```

```

  param: string;
  rez: integer;
end;

```

```

begin
  readln(n); a := '1';
  for i := 1 to n do begin
    readln(s);
    pos(a, s) Mass[i].param := copy(s, 1, pos(a, s) - 1);
    delete(s, 1, pos(a, s));
    val(s, Mass[i].rez, kod);
  end;

```

```

  for i := 1 to n do begin
    max := Mass[i].rez; i_max := i;
    for k := i to n do if Mass[k].rez > max then max := Mass[k].rez;
      begin
        max := Mass[k].rez;
        i_max := k;
      end;
  end;

```



№3 - нет



```

b := Mass[i].rez Mass[i].rez; c := Mass[i].param;
Mass[i].rez := Mass[imax].rez; Mass[i].param := Mass[imax].param;
Mass[imax].rez := b; Mass[imax].param := c;

```

```
end;
```

```
for i := 1 to n do writeln(i, ' ', Mass[i].param, ' ', Mass[i].rez);
end.
```

N=1

```
var a, b, i: integer;
```

```
function power step(a, n: longint): longint;
var f, i: longint;
```

```
begin
f := 1;
for i := 1 to n do f := f * a;
end step := f;
end;
```

```
begin
```

```
readln(a);
```

```
k := 0; b := 0
```

```
while a > 0 do begin
b := b + (a mod 10) * step(8, k);
k := k + 1;
a := a div 10;
end;
```

```
for i := 1 to b div 1000 do write('M');
```

```
for b := b mod 1000;
```

```
for i := 1 to b div 500 do write('O');
```

```
b := b mod 500;
```

```
for i := 1 to b div 100 do write('C');
```

```
for b := b mod 100;
```

```
for i := 1 to b div 50 do write('L');
```

```
b := b mod 50;
```

```
for i := 1 to b div 10 do write('X');
```

```
b := b mod 10;
```

```
for i := 1 to b div 5 do write('V');
```

```
for b := b mod 5;
```

```
if b > 1 then write('I');
```

```
for i := 1 to b mod 1 do write('I');
```

```
end.
```

(F)

N=6var mass1, mass2: array [1..100] of integeri, s1, s2, n: integerbegin

readln(n);

for i:=1 to n do readln(mass1[i], mass2[i]); s1:=0; s2:=0

for i:=1 to n do begin

s1:=mass1[i]+s1;

s2:=mass2[i]+s2;

end;

if s1>s2 then writeln(s2);else writeln(s1);

end.

N=2Когда нам нужен доступ к члену, и, чтобы не переводить
из 2cc в 10cc ~~и~~ обратно, используя глобально-звоня. с.с.

Выдан дополнительный лист: 1

Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7102

ФАМИЛИЯ МИХАЙЛОВА

ИМЯ АННА

ОТЧЕСТВО ВЛАДИМИРОВНА

Дата рождения 18.07.1998

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:

Михайлова

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



```

Var ① Mass: array [1..10 1..10 1..10 1..10 1..10 1..10 1..10 1..10 1..10 1..10] of string;
a, k, b, d, i: integer;
Begin
  Readln (a);
  k := 0; Mass[1] := ''; Mass[2] := ''; Mass[3] := ''; Mass[4] := '';
  while a <> 0 do
    Begin
      b := (a mod 10) * 8^k);
      a := a div 10;
      inc (k);
    end;
    if b > 3999 then Writeln ('not founded')
    else Begin
      d := b mod 10
      if d = 1 then Mass[1] := 'I';
      if d = 2 then Mass[1] := 'II';
      if d = 3 then Mass[1] := 'III';
      if d = 4 then Mass[1] := 'IV';
      if d = 5 then Mass[1] := 'V';
      if d = 6 then Mass[1] := 'VI';
      if d = 7 then Mass[1] := 'VII';
      if d = 8 then Mass[1] := 'VIII';
      if d = 9 then Mass[1] := 'IX';
      if d =
      b := (b div 10); if b > 0 then Begin
        d := (b mod 10) * 10;
        if d = 10 then Mass[2] := 'X';
        if d = 20 then Mass[2] := 'XX';
        if d = 30 then Mass[2] := 'XXX';
        if d = 40 then Mass[2] := 'XL';
        if d = 50 then Mass[2] := 'L';
        if d = 60 then Mass[2] := 'LX';
        if d = 70 then Mass[2] := 'LXX';
        if d = 80 then Mass[2] := 'LXXX';
        if d = 90 then Mass[2] := 'XC'; end;
      b := b div 10; if b > 0 then Begin
        d := (b mod 10) * 100;
        if d = 100 then Mass[3] := 'C';
        if d = 200 then Mass[3] := 'CC';
        if d = 300 then Mass[3] := 'CCC';
        if d = 400 then Mass[3] := 'CD';
        if d = 500 then Mass[3] := 'D';

```





```
if d = 600 then Mass[3] := 'DC';
if d = 700 then Mass[3] := 'DCC';
if d = 800 then Mass[3] := 'DCCC';
if d = 900 then Mass[3] := 'CM'; end;
b := b div 10; if b > 0 then Begin
d := (b mod 10) * 1000;
if d = 1000 then Mass[4] := 'M';
if d = 2000 then Mass[4] := 'MM';
if d = 3000 then Mass[4] := 'MMM'; end;
end;
for i := 4 down to 1 do
if Mass[i] <> '' then Write(Mass[i]);
end.
```

②

Применение такой формы записи целесообразно, когда необходимо работать не с числом, а с отдельно взятой цифрой в десятичном числе.

③

```
Var a, b, c: integer;
```

```
Begin
```

```
Read(a, b);
```

```
While a <> b do
```

```
if a > b then a := a - b;
```

```
else b := b - a;
```

```
if a <> 1 then c := a + b
```

```
else c := 0;
```

```
if c <> 0 then Writeln('не существует')
```

```
else Writeln(c);
```

```
end.
```

```
⑤ Var A: array [1..10000] of record
```

```
zn: string;
```

```
vez: integer;
```

```
end;
```

```
i, max, k, N: integer;
```




```
Begin
  Readln (N);
  for i := 1 to N do
  Begin
    Readln (Mass[i].zn);
    Read (Mass[i].vez);
  end;
  Writeln;
  max := zn Mass[1].zn; k := 1;
  While k <= N do
  Begin
    for i := 1 to N do
      if Mass[i].zn > max then max := Mass[i].zn;
    for i := 1 to N do Begin
      if Mass[i].zn = max then
        Writeln (Mass[i].zn, '-', Mass[i].vez);
        Mass[i].zn := 0; end;
    inc (k);
  end;
end.
```



④ Var Mass: array [1..15000] of integer; i, k, N, S: integer;

```
Begin
  Readln (N);
  Mass[1] := 4; Mass[2] := 3; S := 0;
  for i := 3 to N do Mass[i] := 2;
  k := 2; i := 3;
  While i <= N do Begin
    Mass [i+k] := 3;
    Mass [i+2*k] := 3;
    i := i+2*k; k := k+1;
  end;
```



```
for i := 1 to N do
  S := S + Mass[i]; Write (S);
end.
```



⑥

Var Mass: array [1..100, 1..100] of integer;

M, N, i, j, S, S1: integer;

Begin

Readln(M, N);



for i := 1 to M do

for j := 1 to N do

Read (Mass[i, j]); S := 0; S1 := 0;

j := 2;

While j <= N do Begin

for i := 1 to M do S := S + Mass[i, j];

j := j + 3; end;

j := 3

While j <= N do Begin

for i := 1 to M do S1 := S1 + Mass[i, j];

j := j + 3; end;

if S > S1 then Write ('Temp')

else Write ('Damm');

end.

if S > S1 then Write ('Temp')

else if S = S1 then Write ('ниче')

else Write ('Damm');

end.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7092

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Николаев

ИМЯ Степан

ОТЧЕСТВО Александрович

Дата рождения 08.09.1999

Класс: 9

Предмет информатика

Этап: заключительный

Работа выполнена на 3 листах

Дата выполнения работы: 17.01.15
(число, месяц, год)

Подпись участника олимпиады:

ИЧ

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



```

4. program z7;
var a: array [1..100000] of integer; l, n, i, s, j: longint;
begin
a[1] := 1;
l := 1;
n := 0;
for i := 2 to 44 do
begin
s := l;
for j := 1 to s do
begin
a[j] := a[j] + n;
m := a[j] * i;
a[j] := m mod 10;
n := m div 10;
if a[j+1] = 0 then a[j+1] := n;
if a[j+1] > < 0 then l := l + 1;
end;
end;
if a[j+1] > < 0 then s := j + 1 else s := j;
for i := s downto 1 do
if a[i] = 4 then write(i, ' ');
readln;
end.

```

вычисляем факториал 44
в массиве.



определяем номера
разрядов, содержащих
цифру 4.

" " - означает пробел.

```

5. program z5;
var l: longint;
begin
l := 10000 div 44;
l := (l + 1) * 44;
writeln(l);
while (l <= 99999999) do
begin
l := l + 44;
writeln(l);
end;
readln;
end.

```

находим первое число, которое : 4 и : 11, является
кратным числом. Затем сразу же его выводим.

Далее выводим числа из промежутка
 $10000 \leq x \leq 99999999$, при этом числа,
которые выводим, кратны 4 и 11.





```

3. program z3;
var n, i, j, s, l: longint;
begin
  readln(n); l:=0; if (n<>1) and (n<>2) then
    for i:=2 to n-1 do
      begin
        s:=0;
        for j:=2 to i
          if (n mod j=0) and (i mod j=0) then s:=s+1;
          if s=0 then l:=l+1;
        end else l:=0;
      writeLn(l+1);
    readln;
  end.

```

То есть, мы находим такие числа от 1 до N, число N не делится на эти числа. В начале программы я не указываю 1, т.к. оно подходит для любого числа.

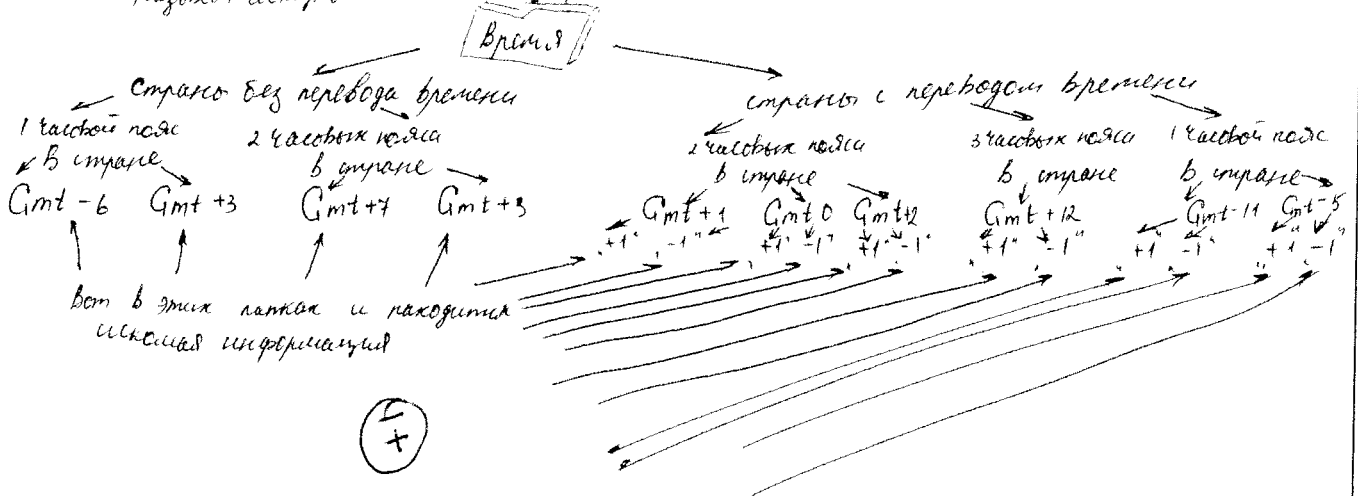


2. Сначала следует разбить все страны на два каталога: с переводом времени и без перевода времени.

Рассмотрим группу стран без перевода времени. Этот каталог разобьем на количество часовых поясов в стране. Далее разбиваем каждый каталог с количеством часовых поясов в стране на часовые пояса, в которых будет указано местное время в заданной стране.

Вернемся к группе стран с переводом времени. Аналогично разбиваем этот каталог на группы по количеству часовых поясов в стране. Сейчас представим часовые пояса этих стран как бы время не было со переводом. Разбиваем каталоги по количеству часовых поясов в стране на часовые пояса, которые мы сейчас представим. В каждом из полученных каталогов создаем еще по два подкаталога: "+1" и "-1" (это означает перевод времени). В этих группах и будет храниться информация о местном времени в указанной стране.

Представим все это в иерархической системе (для примера) Назовем исходный каталог "Время".





4. 1) Запишет все числа в массив. { например число 1632871.
 2) Начнет с конца массива брать из второй строки ~~цифры~~ цифру, учитывая разрядность. Во второй строке то же самое число
 В нашем примере число 1632871 сначала умножает на 1 (по цифре), затем на 4, 8, 2, 3, 6, 1. (не забудете учитывать разрядность!)
 Делает эту операцию, как при вычислении факториала в 4 задаче.

1	1	6	3	2	8	4	1
1	1	6	3	2	8	7	1

Во второй строке то же самое число

Делает эту операцию, как при вычислении факториала в 4 задаче.

$$\begin{aligned}
 m &:= a[j] * a[2, i] \\
 a[j, j] &:= m \bmod 10; \\
 n &:= m \operatorname{div} 10; \\
 \text{if } a[j, j+1] = 0 \text{ then } a[j, j+1] &:= n.
 \end{aligned}$$



Вот так прогоняет весь первый ряд.

Указательно ~~будет~~ перед прогоном каждого ряда

$n=0$. Повторю еще раз, что начинаем

прогонять ряд слева направо.

Угол первого ряда означает (в нашем случае)

первую строку умножает сначала на 1, потом на 4, 8, 2, 3, 6, 1 (учитывать разрядность!)

$a[2, i]$ принимает переиспользуемые значения.

Делает эти прогоны M раз.

Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7113

ФАМИЛИЯ НИСТРАТОВ

ИМЯ ВАСИЛИЙ

ОТЧЕСТВО АНДРЕЕВИЧ.

Дата рождения 10.05.1997

Класс: 11

Предмет ИНФОРМАТИКА.

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ.

Работа выполнена на 5 листах

Дата выполнения работы: 15.03.2015.
(число, месяц, год)

Подпись участника олимпиады: _____

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



Алгоритм
22

Итак размер поля известен, то будем использовать массив.
 размером $[\frac{K}{2}, \frac{L}{2}]$ (т.к. K и L будут введены, а каждая ель
 занимает не более 2х клет., далее создадим массив
 размером $[N-M+1]$ (который будет отвечать за кол-во
 деревьев елей на поле.) Далее будем заполнять массив
 поля, сразу же добавляя по единице, если встретим
 ель в ячейку = ярусам ели. Затем начинаем от 1 до
 $(N-M-1)$, будем печатать: "Ярус" номер ячейки
 и её значение, если значение ячейки $\neq 0$.

Алгоритм:

Объявим целочисленные переменные $K; L; N; M; i; j;$

Считаем значения $K; L; M; N$

Создадим массив $Mass[\frac{K}{2}, \frac{L}{2}]$ (окружная земля

в большую сторону, целый раз.)

Создадим массив $Знач[N-M+1]$.

Заполним $Mass$ циклом i от 1 до $\frac{K}{2}$

в цикле ~~каждый~~ каждый циклы j от 1 до $\frac{L}{2}$,

если на поле в ячейке $[i; j]$ пусто, то

$$Mass[i; j] = 0$$

иначе

$$Mass[i; j] = \text{ярус дерева}$$

$$Знач[\text{ярус дерева}] = Знач[\text{ярус дерева}] + 1.$$

концы ели.

концы цикла

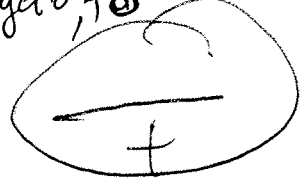
концы цикла

циклы i от 1 до $(N-M+1)$

если $Знач[i] < > 0$ то

вывести: "Ярус: " + i + " кол-во: " + $Знач[i]$

концы ели
концы поля.





№5 Создадим массив размером $[M; N]$, затем считаем строки текста с страницы блокнота, затем для уменьшения кол-ва работы, посчитаем кол-во строк, ^{первый} символов, которых \neq " " (не пустой), заменим число строк Z переменной "Q", затем заменим все строки разделив их " " (пробелом) в одну переменную. типа строки: создадим список и будем добавлять в конец значение каждый раз, когда будем считать в строке новое слово, законим цикл, когда после цифри, будет два пропуски знака (пробела). Заменим кол-во строк в списке в переменную "Z". Пройдем циклом от 1 до $Z-1$ и в нем циклом от 1 до $Z-1$ отсортируем по убыванию (сравнение текущий и следующий элемент списка), затем в массив уменьшим элемент.

Затем циклом от 1 до Z найдем все значения массива.

АЛГОРИТМ:

Создадим уменьшенные переменные: $Z; M; N; i; Y; K$

Создадим переменную типа: строка: Q

Считаем $M; N$.

Создадим массив $Mass [M; N]$.

Считаем строки блокнота. в $Mass [M; N]$.

Циклом i : от 1 до M

Найдем кол-во строк, где первый символ \neq " "

концу цикла.

Заменим полученное значение в "Y".

~~Заменим в строке с " " пробелом~~

~~и " "~~

циклом i : от 1 до ~~Y~~ N

циклом, j : от 1 до $Y-1$

Если $(j \text{ символ} + k \text{ символ}) < >$ " ", то

~~концу~~ концы внутреннего цикла.
 ~~масс.~~





Если $j \neq k$ и $k \neq 0$ то $k = +1$

Замечаем ~~что~~ в конце списка

имеем

$k = k + 1$ (k позволяет нам знать длину списка

конца списка

конца списка

конца цикла

конца цикла.

Создадим массив z с k элементами $z[i] = \text{число элементов в списке}$
циклом $i = 0$ от 1 до k масса элементов в списке
перенесем значения $z[i] = i$ списка в массив

конца цикла $k = 0$;
 $z = \text{масса элементов в списке}$

циклом $i = 0$ от 1 до $z - 1$

циклом $j = 0$ от 1 до $z - 1$

Если $z[j] < z[j + 1]$ то

$k = z[j + 1]$

$z[j + 1] = z[j]$

$z[j] = k$.

конца цикла

конца цикла

конца цикла

Отразим все числа по убыванию, используя цикл

цикл $i = 0$ от 1 до z

Написать в строку $z[z]$

конца цикла.

N 1. Сначала рассмотрим число в переменной "Q"
Затем будем сравнивать Q с числами Фибоначчи

пока не найдем то, которое больше Q. Запомним

(номер этого числа - 1) в переменной N, затем
создадим массив (~~функцию~~) с элементом [N], ~~и~~

это будет вычитать из Q число Фибоначчи под

номером iz от N до 1 (изначально посчитали и записали

и числа Фибоначчи в массив. Znach)



Если $Q > \text{Знач}[i]$, то будем заносить в ячейку массива 1 число 0. Таким образом заполним массив 1 и 0, отразим его на экране.

Алгоритм:

Создадим умноженные переменные: $N; Y_1; Y_2; Q; Y; Z$

Считаем $Q; Y=1; Y_1=1; Y_2=1$

Пока $Q > Y$

$$N = N + 1$$

$$Z = Y$$

$$Y = Y_2 + Y_1$$

$$Y_1 = Y_2$$

$$Y_2 = Z$$

Конец пока; $Y=1; Y_1=1; Y_2=1; N=N-1$; создадим массив $\text{Mass}[N]$

циклом i от 1 до N

$\text{Mass}[i] = Y$

$$Y = Y_2 + Y_1$$

$$Y_1 = Y_2$$

$$Y_2 = Z$$

Конец цикла. создадим массив $\text{Знач}[N]$.
циклом i от 1 до N

Если $Q \geq \text{Mass}[N-i+1]$, то

$$Q = Q - \text{Mass}[N-i+1]$$

$$\text{Знач}[i] = 1$$

иначе

$$\text{Знач}[i] = 0$$

Конец если.

Конец цикла. циклом i от 1 до N

напишем в строку значение массива $\text{Знач}[i]$.

Конец цикла.

НУ III.к. Число N^m может выскочить за размеры умноженной переменной, но можно





использовать списки, т.к. кол-во цифр ~~не~~ получившая
 масса не известно. Тогда можно представить каждую
 цифру, как новую ячейку списка и вращать циклом,
 умножать N на N, M раз. Второй способ
 заключается в том, что можно взять
~~умноженный~~ массив, типа `long` и умножать
 N на N, M раз, до того момента, пока
 это не превысит значение (максимальное) типа `long`,
 затем перейти в функцию ячейку массива с соответ-
 ствующими переносами чисел. Третий способ заключается
 в разбиении N^m на $(N^2)^m$ и дальнейшей
 ручной работой в текстовой переменной (система вращает
 через ^{оператор} "еще!")

(+)

N2. Создадим два массива, размером ~~элементов~~ = всем комбинациям
 набору из четырех чисел в промежутке от 1 до 50.
 В первый массив запишем все числа (считаем их
 и переберем варианты один четырех квадратов чисел,
 а во второй будем записывать в первый столбец
 число, во второй кол-во раз, которое оно повторилось.
 Затем будем искать числа, которые встречаются,
 при переборе массива из первого столбца массива
 и сам значение второго столбца в той же строке,
 меньше переменной, которая считала ~~минимальное~~
 кол-во представлений, то запишем в нее значение
~~второго~~ второго столбца, а в функцию передадим значение
 первого столбца той же строки. Тем самым мы
 найдем необходимые числа N .

(+)

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7102

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ПУГАЧЕВА

ИМЯ ИРИНА

ОТЧЕСТВО ДМИТРИЕВНА

Дата рождения 18.08.1998.

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:

Ирина

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



ЗАДАНИЕ 1.

визит 1 стр. лист

Длина чисел, записанных в римской системе счисления, велика; такое предположение не используется в каких-либо серьезных целях \Rightarrow вводимые числа вряд ли будут превышать число, являющееся максимальным в самом широком диапазоне выбранного языка программирования (в Паскале $\approx 10^{18}$) \Rightarrow десятичные числа, равные восьмеричным, можно хранить как числа (сохранять их в переменную целого типа).

Разобьем поставленную задачу на 2 подзадачи: 1) перевод данного восьмеричного числа в десятичную систему счисления; 2) представление десятичного числа в римской системе счисления.

Решение первой подзадачи (+ считывание исходных данных): понадобятся три переменные: 1) a : тип данных максимально возможной (в Паскале int64 , например); 2) n : тип данных такой, что можно считать цифру (в Паскале byte); 3) c : символ.

\Rightarrow ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ:

```

read(c); if c = '-' then read(a) else val(c, a, n);
while not (eoln) do
begin
  read(n);
  a := a * 8 + n;
end;

```

Если $c = '-'$, то после ввода нужно вывести 1 , т.к. этот символ будет стоять в начале римского числа.



Итог работы данного фрагмента: в переменной a хранится исходное восьмеричное число, переведенное в десятичную систему счисления.

Решение второй подзадачи:

Возьмем новую переменную kt (количество тысяч), присвоим ей значение $kt := a \text{ div } 1000$. В начале нашего римского числа будет стоять kt букв "M". Можно сразу их вывести:

\Rightarrow ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ:

```

for n := 1 to kt do
  write("M");

```

Пояснение: переменная n использовалась при счете, но к этому моменту она уже не нужна; если в числе нет тысяч (десятичное число получилось меньше 1000), ничего выводиться не будет.

Тысячи числа a нас уже не интересуют, выполним операцию $a := a \text{ mod } 1000$. Сохраним число a как массив $m[1..3]$, где $m[1]$ - количество сотен, $m[2]$ - количество десятков, $m[3]$ - количество единиц.

\Rightarrow ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ:

```

m[1] := a div 100;
m[2] := (a div 10) mod 10;
m[3] := a mod 10;

```

Создадим массив $v[1..3, 1..3]$, такой, что

$v[1,1] = C$	$v[1,2] = D$	$v[1,3] = M$
$v[2,1] = X$	$v[2,2] = L$	$v[2,3] = C$
$v[3,1] = I$	$v[3,2] = V$	$v[3,3] = X$

Далее основная часть второй подзадачи:

\Rightarrow ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ

```

for n := 1 to 3 do

```

```

begin
  case m[n] of
    1: write(v[n,1]);
    2: write(v[n,1], v[n,1]);
    3: write(v[n,1], v[n,1], v[n,1]);
    4: write(v[n,1], v[n,2]);
    5: write(v[n,2]);
    6: write(v[n,2], v[n,1]);
    7: write(v[n,2], v[n,1], v[n,1]);
    8: write(v[n,2], v[n,1], v[n,1], v[n,1]);
    9: write(v[n,1], v[n,3]);
  end;
end;

```

Отдельно до этих строк в программе следует рассмотреть случаи, когда a - «особое число». Например, если $a = 99$, то оно записывается как IC, а не как XCIX. Т.е. программа будет выглядеть примерно так:

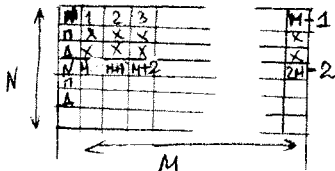
Если a - особое число, то $\{ \}$, иначе фрагмент программы выше.



Учитывая, что «особых» чисел ограниченное количество, можно в начале программы создать массив соответствий: «особое» число — его римская запись. Тогда на месте слов «если a — особое число...» будет проход по этому массиву.

Задача 6.

Допустим, мальчики записывали результаты игры горизонтально, при этом они хотели записать результаты как можно большего числа партий, поэтому их листок выглядел примерно так:



Т.е. они начинали записывать результаты с левого верхнего угла, заканчивая 3 строки и горизонтально могли записать $(M-1)$ игры, затем они переходили на строки 4-6 и т.д. При этом вертикально могло использоваться $(N \div 3) * 3$ строк, чтобы для каждой группы игр имелись обозначения N° и П. Д. (Если N при делении на 3 дает в остатке 2, то вообще может быть записано N° и П, а результатам Данила не хватит места). При этом M — любое число, большее 1, чтобы этот листок вообще нес какую-либо информацию. Пусть каждая клетка соответствует ровно одному результату, пустые клетки могут присутствовать только в конце, когда кончатся игры. При вводе эти пустые клетки обозначаются пробелами.

Вначале просто считаем блокнотный лист как массив символов (двумерный) $M \times N$.

В переменную s сохраняем количество трехстрочных блоков ($s = N \div 3$). Посчитаем две нужные нам суммы, пройдясь по массиву.

⇒ ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ:

```
sp:=0; sd:=0; for i:=1 to s do
  for j:=2 to M do
    begin
      val(mas[i*3-1, j], rp, c);
      val(mas[i*3, j], rd, c);
      sp:=sp+rp;
      sd:=sd+rd;
    end;
  end;
```

Используемые переменные:

- 1) i, j, sp, rp, sd, rd — целочисленные переменные типа `integer` или `int64`;
- 2) mas — тот самый двумерный массив, который мы считывали;
- 3) c — переменная типа `integer`.

В результате работы этого фрагмента программы в переменной sp будет сумма Петинских очков, в переменной sd — сумма баллов Данила. Останется только сравнить эти числа и вывести имя школьника, набравшего меньше очков.

⇒ ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ:

```
if sp < sd then writeln('Петя') else writeln('Данил');
```

Задача 5

Количество строк входного файла не известно ⇒ сделаем программу «пока не конец файла».

Считываем построчно (в строке изначально конец, значение, результат). До первого пробела — номер, он нам не нужен, его куда не сохраняем. Далее все до позиции пробела включительно. Далее до следующего пробела (не включительно) идет значение параметра, выделяем эту часть так же с помощью функции поиска позиции, сохраняем в массив $name[i]$ из строк. Удаляем из строки все до позиции пробела включительно там остается только результат, но он представлен как строка, а не как число. Преобразуем его в число с помощью функции val (на языке Паскаль) или т.п. и сохраняем в массив $res[i]$ из чисел (следует учесть, что результаты могут быть большими, поэтому нужно выбрать большой тип данных). Так же при вводе считаем количество введенных строк (N). Далее расположим результаты в порядке убывания (от большего к меньшему), не забывая, что «результат» привязан к значению параметра.

⇒ ФРАГМЕНТ ПРОГРАММЫ НА ЯЗЫКЕ ПАСКАЛЬ



```

for i := 1 to N-1 do
  for j := i+1 to N do
    begin
      if res[i] < res[j] then begin
        k := res[i];
        res[i] := res[j];
        res[j] := k;
        s := name[i];
        name[i] := name[j];
        name[j] := s;
      end;
    end;
  } k, i, j - переменные целого типа;
  s - переменная типа string;

```

В конце от 1 до N нужно вывести:

- 1) номер строки
- 2) name [номер строки]
- 3) res [номер строки]

Вывод будет происходить в порядке убывания, что требовалось в задании.

⇒ фрагмент программы на языке Паскаль:

```

for i := 1 to N do
  write (i, " ", name[i], " ", res[i]);

```

Номеру i будет соответствовать элемент, результат которого находится на i -том месте по убыванию, а не тот, который проводился i -тым по счету. Если нужно сохранить номер элемента покажем массив $n[i]$, при отпывании в него будет сохраняться все до первого пробела, при сортировке с ним будет происходить ровно то же самое, что и с массивом $name[i]$.

Задача 4

Если пытаться ДЕЛАТЬ ТО ЖЕ, ЧТО ДЕЛАЕТ РОБОТ, ПРИ ЭТОМ КАЖДЫЙ РАЗ ЗАМЕНИВ НЕ ОБХОДИМО КОЛИЧЕСТВО РЕК ДЛЯ НОВОГО КВАДРАТА, ТО МОЖНО ЗАМЕТИТЬ ЗАКОНОМЕРНОСТЬ:

4 3 3; 2 3 2 3 2 2 3 2 2 2 3 2 2 2 2 3 2 2 2 2 3 2 2 2 2 2 3 ...

1
1
2
2
3
3
4
4
5

Это легко объяснить тем, что сторона квадрата постепенно увеличивается. Каждая цифра ряда выше означает, сколько рек нужно для n -ного квадрата, а сумма первых n чисел означает, сколько рек нужно всего для N квадратов.

Возьмем переменную j - номер квадрата (или количество квадратов).

```

i := 0; while i < 15000 do
  j := 2; begin
    i := i + j;
    a[i] := 3;
    i := i + j;
    a[i] := 3;
    inc(j);
  end;

```

В результате работы данного фрагмента программы у нас будет массив, в котором на n -тых местах стоят тройки (только до этого массив нужно обнулить). Далее $a[1]$ присвоим значение 4, $a[2] = 3$, $a[3] = 3$.

Проходим по массиву еще раз (от 1 до 15001); если $a[i]$ равно 0, т.е. в первый раз оно не изменилось, то изменяем значение $a[i]$ на 2. Это всё происходит в программе каждый раз ещё до считывания N (операций не так много, пользователь не заметит задержки). После считывания N считается сумма первых N элементов и она выводится на экран. Это и будет искомым количеством рек.

Задача 3

- После считывания n и a узнаем, являются ли они взаимно простыми:
- найдем меньшее из них операцией сравнения
 - от 2 до меньшего в цикле пытаемся разделить n и a на i - переменную - счетчик цикла. Если они оба делятся, то увеличиваем целочисленную переменную s на единицу (или переменную типа boolean делаем равной true - так меньше затрачивается памяти)
 - после цикла смотрим на значение s (пусть она типа boolean, в начале программы была false).
Если ее значение false, то не нашлось такого i , на которое бы делили n и a , и $n \Rightarrow \text{НОД}(a, n) = 1$.
Если ее значение true, то существует i , $i \in [2; \min(a, n)]$, такое, что $a; i$ и $n; i$ одновременно $\Rightarrow \text{НОД}(a, n) \neq 1$.
 - Вспомним условие: мультипликативно обратное число существует тогда и только тогда, когда $\text{НОД}(a, n) = 1$. Т.е. если наша переменная s в результате работы предшествующей части программы меняла значение на true (нашлось $\text{НОД}(a, n) > 1$), то результат работы всей программы - сообщение «НЕ СУЩЕСТВУЕТ!». Иначе происходит выполнение п. 5.
 - $\text{НОД}(a, n) = 1 \Rightarrow$ мультипликативно простое число существует.

»фрагмент программы на языке Паскаль:

```
t:=false; while t<>true do
  i:=1;
  begin
    if a*i mod n = 1 then t:=true
    else inc(i);
  end; <<
```

{t - еще одна переменная типа boolean, i - целочисленная переменная, которая еще была счетчиком}

Когда цикл завершится, переменная i будет иметь значение, которое нам нужно найти (i - мультипликативно обратное число для числа a)
Останется только вывести i .

(Структура программы будет следующей:
(описание переменных)
начало

если a и n - взаимно просты, то цикл по поиску i
иначе вывод «не существует»
конец.)

Задача 2

Использование двоично-десятичной системы счисления целовообразно при

- шифровании (простой способ, ненадежный, но лучше, чем ничего)
- если в компьютере нет АЛУ, но как-то заложена таблица соответствия цифр десятичных и их двоичных кодов (с помощью электрических импульсов могут быть переданы числовые данные)
- если числа, с которыми работает программа, очень большие (больше, чем $\approx 2^{16}$), то запись в двоичной форме будет занимать не наименьшее количество мест, чем в двоично-десятичной (большее количество символов), ~~так при увеличении числа в два раза в двоичной системе только добавляется цифра, а в десятичной~~

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ДАЧКОВА

ИМЯ АНАСТАСИЯ

ОТЧЕСТВО СЕРГЕЕВНА

Дата рождения 07.07.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 04 листах

Дата выполнения работы: 15.03.15
(число, месяц, год)

Подпись участника олимпиады: _____



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№1 a - исходное число в десятичной системе
 k, i - счётчик $a[i]$ - массив числами Фибоначчи
 0 - минимальная величина v - величина переменной

Начало ~~программы~~ алгоритма:
 Введи число a ;
 $i := 1$;

Пока $v \neq 0$
 начало цикла

$v := a : 10$;

Берём целую часть от v ;
 $k := k + 1$;

конец цикла;

$a[1] := 1$;
 $a[2] := 2$;

~~Начало цикла~~

Для j от 3 до i

начало цикла
 Пока $a[j] < i$

начало цикла

$a[j] := a[j-1] + a[j-2]$

конец цикла;
 конец цикла;

Для j от k до 1 (отчёт от большего
 к меньшему)
 начало цикла

Если $v - a[j] > 0$, то $v := v - a[j]$;
 v приписывает справа "1"

иначе v приписывает справа "0"

0 - ответ к задаче.

Конец алгоритма.

№2 $b[i]$ - массив с вариантами представления
 $a[i]$ - массив с квадратами чисел
 начало алгоритма

Введи N ;

$k := 0$;

Для j от N до $N+50$

начало цикла

$v := j$;

считаем v

~~конец цикла~~

→ см. др. стр.

считаем
 кол-во
 разрядов в
 числе

ищем
 числа
 Фибоначчи
 и
 сохраняем
 в массив





Для $i := \text{от } 1 \text{ до } b$

$$a[i] := i^2$$

находим количество комбинаций (переменная k), из которых можно составить число j

$$b[k] := j$$

Если $k_{\text{min}} > k$, то $k_{\text{min}} := k$

Если $b[k] < 1$, то $b[k] := 0$

конец цикла;

Конец алгоритма.

Таким образом, в самом начале массива будут находиться числа, которые можно представить в виде суммы четырёх квадратов натуральных чисел наименьшим количеством слагаемых.

n 3 p -исходное число

$$p1 = p - 1$$

Начало алгоритма

$$b := 1;$$

$$s := 0;$$

Выводим число p ;

$$p1 := p - 1;$$

$$b := p1;$$

Пока остаток от $p1$ не равен нулю

начало цикла

$$s := s + 1$$

конец цикла

$$d := p1 : 2^s;$$

$$t := \log_2(p)$$

Определим t

Для $i := \text{от } 1 \text{ до } t$

начало цикла

выбираем случайное ~~число~~ ^{a} из диапазона $[1, p1]$

если $a^d = 1$, то выводим и веримто простое?

СМ. ДР. СБР. →





иначе выбираем случайное k из диапазона $[0; s-1]$

$$k = a^k$$

если $a^{k \cdot d} = -1$, то выводим «вероятно простое»

иначе выводим «составное»;

конец цикла;

конец алгоритма.

№4 отн - ответ

начало алгоритма


Введем M ;

введем N ;

если $(M < 10^{1000})$ и $(N < 10^{1000})$, то

Для $i :=$ от 1 до M

$$\text{отн} := \text{отн} \cdot N$$

конец алгоритма. 

№5 $a[i, j]$ - двумерный массив с элементами, которые записаны в строке i и столбце j

По условию значения $a[i, j]$ равны.

начало алгоритма

Для $i :=$ от 1 до $N-1$

Для $j :=$ от 1 до $M-1$

Если $a[i, j] < a[i, j+1]$

начало цикла

Для $i :=$ от 1 до $N-1$

Для $j :=$ от 1 до $M-1$

начало цикла

если $a[i, j] < a[i, j+1]$, то $b := a[i, j]$;
 $a[i, j] := a[i, j+1]$;
 $a[i, j+1] := b$;

конец цикла;

конец алгоритма.

СМ. ДР. СТР. →



№ 6 Массив $a[i]$, где $i \in [1; N]$

массив $b[r, j]$ - двумерный, где $r \in [1; \frac{K}{2}]$

vet - количество узлов веток

$j \in [1; \frac{L}{2}]$

$b[r, j]$ - ячейка массива $b[r, j]$ - квадрат с предположением

Начало алгоритма

Для $r := 05$ 1 до $\frac{K}{2}$

Для $j := 05$ 1 до $\frac{L}{2}$

если на $a[r, j]$ есть ель, то $a[r, j] = 1$
иначе $a[r, j] = 0$

Для $r := 05$ 1 до $\frac{K}{2}$

Для $j := 05$ 1 до $\frac{L}{2}$

если $b[r, j] \neq 0$

то $a[vet+1] := a[vet] + 1$
конец алгоритма.

Таким образом, данные о количестве елей
каждой клетки будут выглядеть так:

...	4	5	6	7	8	...
	3	2	0	1	5	

↑ количество елей веток
↑ количество елей ранков вета.

Данные о расположении елей выглядят так:

	0	1	1
к	1	0	0
	1	1	0

↑ есть ель
↑ нет елей

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ РОМАШКИНА

ИМЯ СВЕТЛАНА

ОТЧЕСТВО АЛЕКСЕЕВНА

Дата рождения 29.05.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: _____

Работа выполнена на 2 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады:

Ромашкина

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



1) а) 2 переменных для массива, 1 пер. для перевернутого числа, ~~каждый элемент массива~~ ^{по 2} массив.
 б) Вычисляются числа ~~перевернутого~~ (с сохр. пер.) Приваливание поперек.
 если у нас 2 пер., то четные номера хранятся в \bar{u} , нечетные в \bar{l}
 + сохраняется номер перевернутого ~~э-та~~
 в) как только получившееся число больше искомого, вычитается перевернутое.
 В ~~массиве~~ массиве ~~на~~ ^{в э-те} месте номером внешнего числа в ряду, ставится ~~1~~

в) повторяется до победного ($N=0$).

б) Бинарный массив, p э-тов. посл. обходится участок, вносящие поперек
 ко все встретившиеся кол-ва э-тосов. на оставшихся места
 (г.к. не все сажены есть сейчас) ставятся "0".
 Массив упорядочивается любым способом, после чего подчитывать
 число э-тов заданного значения можно даже ~~место~~ ^{вызвано}
 можно не упорядочивать, а просто подчитывать посл. кол-во э-тов с
 значениями $= M, M+1, \dots, N - 0$ (обычным поэлементным
 сравнением).

5) • Все числа вносятся в массив
 • Ищется максимум
 • меняется с 2 э-том
 • Повторяем с второго э-та или 1-3 пока поиск не будет найден
 с $N-1$ э-та, где N - ~~число~~ ^{кол-во} чисел на бумаге

3) $p-1$ посл. делим на 2 до получения нечетного числа (находим 4 и 5)

```

• находим  $t := \log_2 p$ 
•  $\text{randomize}$ ;
• Для  $i$  от 1 до  $t$ 
   $u := \text{random}(p-2) + 1$ ;
  если  $\exp(d \cdot \ln a) \bmod p \neq 1$  то
     $\exp(d \cdot \ln a) \bmod p = 1$ 
     $d := d + 1$ ;  $r := 0$ ;
  пока  $p \leq u$  и  $r \leq (s-1)$ 
  если  $\exp(\exp(r \cdot \ln 2) \cdot d \cdot \ln a) \bmod p = 1$  то  $fl := \text{false}$ 
  иначе иначе  $r := r + 1$ ;
  если  $fl = \text{true}$  то вывод "составное"; halt;
  кв вывод "вероятно простое"
  
```



Обозначения соответствуют условию задачи.
 4) Если в нашем распоряжении ~~идеальный компьютер из будущего~~, то вывод $\exp(M \ln N)$
 Иначе: • Разбиваем число N на 2 слагаемых, равных если четное и посл. если нечет.
 (N_1 и N_2)
 • По возможности, разбиваем M на слагаемые
 • Разложим в большую u степень по формуле

$$N_2^M = C_{M_1}^M \cdot \exp(M_1 \ln N_1) + C_{M_2}^M \cdot \exp(M_2 \ln N_1) \cdot N_2 + C_{M_3}^M \cdot \exp(M_3 \ln N_1) \cdot \exp(2 \ln N_2) + \dots$$

$$+ C_{M_4}^{M-2} \cdot \exp(2 \cdot \ln N_1) \cdot (M-2) \cdot N_2 + C_{M_5}^{M-3} \cdot \exp((M-3) \ln N_2) \cdot N_1 + C_{M_6}^{M-4} \cdot \exp(M \ln N_2)$$

 • Разбиваем на слагаемые
 • повторяем ит 3-4 до победного (ос. M_1 - M_2 -суммируя).





Кемпион и всевозраста:

1) Даны массивы N_1 и N_2 [числа таб A [1:10000], ~~и массивы N_1, N_2~~ ~~и N, R~~]

исх N
рез A

кон числа N_1, N_2, R, y, y_m

Ввод(N);

НЧ N_1

НЧ N_2

если $N = N_1$ то вывод ('1');

если $N = N_2$ то вывод ('0');

~~иначе~~ ~~то~~ ~~иначе~~

~~иначе~~ ~~то~~ ~~иначе~~ $N_1 \neq N_2$ ~~иначе~~ $N_2 \neq N_1$

$R := N$;

иначе $R > 0$;

$N_1 := 1; N_2 := 2; j := 1$

иначе $N_1 \leq R$ и $N_2 \leq R$

если $N_1 > N_2$ то $N_2 := N_1 + N_2$

иначе $N_1 := N_1 + N_2$

$y := y + 1$;

и если $R = N$ то $y_m := j$;

если $N_1 > N_2$ то $A[j] := true; R := R - N_2$;

иначе $A[j] := true; R := R - N_1$;

и

и для j от 1 до ~~кон~~ y_m вывод
если $A[j] = true$ то write(R)
иначе вывод(0)

и

~~и~~ ~~кон~~

5) Даны массивы N [числа таб A [1:10000]]

исх A

рез A

кон числа $y, max, N, dub, j_{max}, t$

~~и~~ ~~вывод~~(~~вывод~~ N)

вывод(N); $max := -1000000$;

для j от 1 до N

и вывод(~~вывод~~ max);

вывод($A[j]$); если $A[j] > max$ то ~~иначе~~ $max := A[j]$;

и $dub := A[j]$

$A[j] := max$;

$A[j_{max}] := dub$;

$t := 2$

для t от 2 до N-1

и $max := -1000000$;

для j от 1 до N если $A[j] > max$ то $y_{max} := j$ и

$dub := A[j]$;

$A[j] := A[j_{max}]$;

$A[j_{max}] := dub$;

и для j от 1 до ~~кон~~ N вывод $A[j]$;

и
кон



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ РОМАШКО
ИМЯ АННА
ОТЧЕСТВО СЕРГЕЕВНА

Дата рождения 04.11.1997

Класс: 11

Предмет информатика

Этап: заключительный

Работа выполнена на 5 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады: ВАС

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



Задача 1

НАЧАЛО

Считать n ; (n - исходное число в 10-ой с.с.)
$$\left. \begin{array}{l} i = 1, k = 1; \\ k \neq 1, m = 0; \\ m \quad i = 1; \end{array} \right\} \text{ - переменные для расчета числа Фибоначчи.}$$
Пока $i \leq n$:

НАЧАЛО

В конец вектора (он изначально пустой) заносим значение i .
$$\left. \begin{array}{l} m = k; \\ k = i; \\ i = m + k; \end{array} \right\} \text{ m и k меняют свои значения и из них формируется следующее число Фибоначчи i}$$

КОНЕЦ

Пока $n < 0$:

НАЧАЛО

Берем число из конца вектора и заносим в переменную i .Если $i \leq n$ тогда

НАЧАЛО

$$r = r * 10 + i; \quad (r \text{ - переменная для хранения числа } n \text{ в представлении с.с. Фибоначчи})$$

$$n = n - \text{число из конца вектора};$$

удалить число из конца вектора.

КОНЕЦ

ИНАЧЕ

НАЧАЛО

$$r = r * 10 + 0;$$

удалить число из конца вектора;

КОНЕЦ

КОНЕЦ

Вывести r .



Задача 5

Листок можно представить в виде 2-мерного массива $M \times N$ клеток.

Тогда нужно выполнить следующие действия:

- 1) упорядочить по убыванию слева направо каждую строку массива
- 2) упорядочить по убыванию сверху вниз каждый столбец массива
- 3) упорядочить по убыванию каждую из диагоналей массива сверху вниз справа налево

Эти действия нужно выполнять до тех пор, пока производится хотя бы одна перестановка.

После этого получится 2-мерный массив, упорядоченный по убыванию слева направо, сверху вниз.

Пример:

Входной:

5	3	11
12	28	4
7	143	15
16	18	2

1)

11	5	3
28	12	4
143	15	7
18	16	2

2)

143	16	7
28	15	4
18	12	3
11	5	2

3)

143	28	18
16	15	12
7	11	5
4	3	2

4)

143	28	18
16	15	12
11	7	5
4	3	2

6)

143	28	18
16	15	12
11	7	5
4	3	2

— КОНЕЦ ПРОГРАММЫ

Примечание: Сортировку можно производить например пузырьковым методом.



Задача 6

Разобьем весь участок эксперимента на квадраты $1\text{ км} \times 1\text{ км}$.

Подсчитаем количество елей каждого вида в каждом из квадратов и занесем это в двумерный массив используя для хранения соответствующий тип данных.

Затем продем по массиву и на каждой шагу прибавляя к общему числу елей данного типа, число елей этого типа в текущей ячейке получим общее число елей на участке.

Подсчет будем выполнять параллельно. Этот алгоритм позволит не только подсчитать число елей ~~в~~ каждого типа в общем, но и получить статистику распределения типов елей по участку эксперимента.

Пример:

Вид участка елей:

1	5	6	5	6
2	7	5	7	5
3	6	7	7	7
4	5	5		
5		6		7

Массив с информацией:

	1	2	3	4
1	5-1 6-0 7-0	5-0 6-1 7-0	5-1 6-0 7-0	6-0 7-1 7-0
2	5-0 6-0 7-1	5-1 6-0 7-1	5-0 6-0 7-0	6-0 7-0 7-0
3	5-0 6-1 7-0	5-0 6-0 7-0	5-0 6-0 7-1	6-1 7-1 7-0
4	5-0 6-0 7-0	6-1 7-0	6-0 7-0	6-0 7-0
5	5-0 6-0 7-0	6-0 7-0	6-0 7-0	6-0 7-1

Число ^в количество елей. Понятно, что ели могут не расти строго в выделенных квадратах. Тогда можно, либо, как представлено выше, записывать ель в тот квадрат, где растет большая её часть, а если эти части равны, то предпочтения отдаётся верхней и левому квадратам.

Второй вариант - это записывать в квадрат дробное значение, соответствующее проценту площади ели данного типа, находящейся в этом квадрате. Такой вариант позволит получить более точную статистику, но общее количество елей от этого не изменится.



Задача 3

Так как числа a выбираются случайно, можно предположить, что случайно выбрано a от 0 до t , чтобы не отнимать одинаковые числа.

$t = \log_2(p)$ с округлением в большую сторону, чтобы не ошибиться.

Все простые числа, кроме 2 и -2 нечетные, так что если вводимое число p четное и не равно 2 и -2, то запускать тест Миллера-Рабина нет смысла и можно просто вывести «Составное» и завершить программу.

Если же число p нечетное (проверится функцией $\text{mod } 2$ на равенство 0), то $p-1$ обязательно будет четным, а все четные числа кратны 2-м.

Тогда можно s присвоить 1, а d присвоить $(p-1)/2$. В случае если p изначально равно 2 или -2, то нужно вывести «Вероятно простое» и завершить программу s нужно присвоить p , а d присвоить $p-1$.

Далее нужно вообразить t как какое-то число и в цикле от $a=1$ до t проверить два условия.

Первое условие можно проверить так: в цикле от ± 1 до $\pm d$ в некоторую переменную m записать m^a . Таким образом в m скажутся a^d .

Далее нужно сравнить полученный результат (из переменной m) с результатом выражения $1 \pm \text{mod } p$. Если они равны, то выводим «Вероятно простое», выводим из цикла и завершаем программу.

В случае если d было отрицательным то в m после цикла нужно записать $\pm 1/m$ и проделать все тоже самое.

Второе условие можно проверить так: Так как s уже равно 1, то r может быть только 0. Тогда достаточно проверить равенство $a^d \equiv -1 \pmod{p}$ по аналогии с первым условием. Если равенство выполняется, то выводим «Вероятно простое» и завершаем программу.

Если ни одно условие не выполнилось ни разу, то можно проверить с помощью переменной r (это можно вывести «Составное» и также завершить программу. Конечно, этот алгоритм не подтвердит в точности теста Миллера-Рабина, однако он не противоречит ему.



Задача 3 продолжение

Если ни одно из условий не выполнилось ни разу (это можно проверить с помощью перемешивания - флагов), то нужно ввести составное и завершить программу.

Конечно, этот алгоритм не повторяет в точности метода Милера-Рабина, однако он не противоречит ему.

Задача 4

Для решения этой задачи нужно использовать длинную арифметику (каждая цифра числа M и N помещается в одну из ячеек массива). Далее можно попытаться найти закон ~~вероятно~~ ^{то} размер массива будет 10^{20000} ~~не~~ так как максимально возможные величины M и N это $1000^{1000} = 1000 (10^3)^{1000} = 10^{3000}$. Вероятно стандартные программы не позволят создать такой массив, так что можно попробовать разбить его на несколько массивов и обрабатывать их вместе.

Потом следует, используя длинную арифметику умножить N на N до тех пор, пока массив с числом, каждый раз умножаемым на единицу, не обнулится.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Рыбин

ИМЯ Станислав

ОТЧЕСТВО Евгеньевич

Дата рождения 14.04.1998

Класс: 11

Предмет Информатика

Этап: Заключительный

Работа выполнена на 5 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады: 

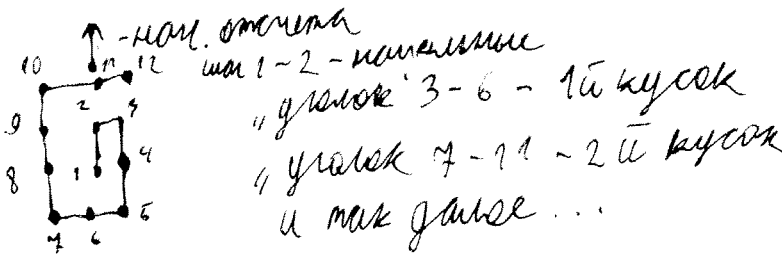
Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



1/4

Всегда год. лист №2 & Вариант год. лист №2 (жест. шорт)

Выделим в двителении работа односторонние участки (кусочек - часть шорта)
Возьмем за начало отсчета второй перестроенный работный квадрат.
Назовем кусочком - отрезок двителения, включающий два поворота.
(Стрелка - траектория работы)



Заметим, что зрейки нуртма в начале кусочка и на
конец кусочка $(m+1)/2$, где m - длина кусочка. Вовсе от-ток - нуртма 2-
кусочек имеет вид $32...332...3$
Заметим алгоритм, составляющий строку S , где N -ый элемент - минимальное кол-во рек для квадрата на шаге N .
Чтобы получить кол-во рек для N квадратов переведем в шала и сложим все элементы строки.

Алгоритм:

- 1) Проверим, что N удов. условием ($N \leq 15000$ и $N \geq 1$), иначе выведем "N не удовлетворяет условию"
- 2) Выведем строку S , $S = "43"$; переменную i , $i = 1$; $ans = 0$
- 3) цикл while, пока выталкивается условие: длина строки меньше N :
приведем к концу строки S строку a , $a = ("3" + "2" \cdot i) \cdot 2$,
где a_i - кусочек номер i , а каждая шорта увели. i на 1.
- 4) Мы получили строку, длины $\geq N$. По условию от 1 до N шорта? (считаем, что первый символ имеет номер 1) приведем к началу ans значение $int(S[1..N])$. ($int(S)$ определяет строку S в шала)
- 5) Выведем ans - ответ на задачу





N3

Алгоритм:

1) Объявим ф-ю $\text{НОУ}(a, b)$, нахождение НОУ : пока a не равно нулю или $b \neq 0$: если $a > b$, то $a = a \% b$, иначе $b = b \% a$. Заметим, что НОУ - циклически не равный шифр - логично.



2) Проверим a и n на удобные условия задачи ($a \geq 0$ и $n > a$ и $n > 0$ и $\text{НОУ}(a, n) = 1$ и a -целое, и n -целое), иначе - выводим ошибку.

4) переим. $k, k = \text{false}$

3) т.к. $b \in [0; n)$ и b -целое, то мы можем перебрать все b целиком: для b от 0 до n проверяем выполняются ли условия

$(a \cdot b - 1) \% n == 0$. Если оно выполняется, то ~~выводим~~ выводим оставшиеся циклы, объявляем переим.-ф-ю k значением true .

4) Если $k = \text{true}$, выводим b , иначе выводим "не существует".

N2

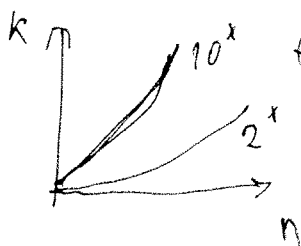
n - число в k -ой системе счисления. В k -ой системе

в 2-ой системе:

$$2^{k-1} \leq n < 2^k - \text{имеет форму } k \text{ знаков}$$

в 10-ой системе:

$$10^{k-1} \leq n < 10^k - k \text{ знаков или } 9 \cdot 10^{k-1}$$



Для 2^x растет медленнее, значит число в k -ой системе растет. Итог - ответ из ее недостатков.



Можно объяснить это подругишу: в 2-ой одинице понравится пере-
счет по е. шил., а можно зафиксировать 76.

Плюсы: очень легко переводить числа из 10-ой в 7-ую:
те же быстрее чем из 2-ой, т.к переводить надо не все число
целое, а каждую цифру по отдельности (можно сделать двойки,
которые нужно выписать = 4, а не к (к-грамма шила), как
в двоичной). \oplus

Плюсы, отражается точность при переводе десятич.
дробей в 7-ую систему. В двоичной точность теряется.
Легко "перевести запятую" - умножить на 10ⁿ.

✓6

считаем, что записывали в одну строку по 3 шила. Строку оди-

н) Проверяем, что длина строки меньше ширины ~~строки~~, что
есть длиннее - полог. шила, иначе выводим вывод об ошибке (шляк исп. формат)

2) Т.к мы не знаем кол-во картон, то будем использо-
вать не массив, а список. Возведем список стр.

3) добавим в спис. стр массив, содержащий результаты всех игр.

4) А = [a, b], где a - рез-т Петри, b - рез-т Дашки. (массив, т.к.
вместе с тем ^{вместе с тем} ~~Петри~~)

5) По ходу игры по списку, каждый шаг прибавляем к результ.

a, к b - значе b.

6) Сравним P и D. Если P > D, то выводим "Петри", если

D > P: выводим "Дашка". Если P = D - "Ничья". \oplus



зад. лист № 1

№5.

Принимая алгоритм сортировки, чтобы использовать его в решении:
н.к. надо быстро отсортировать, использовать алгоритм быстрой сортировки
А-массив

Берем случайный элемент из массива $A[i]$, и поворачиваем по
массиву слева, пока не найдем элемент $A[j]$, что $A[i] \geq A[j]$

и справа-зл-м $A[g]$: $A[i] \leq A[g]$. Поменяем $A[i]$ и $A[g]$
местами. Если та же в обоих эл-ментах получим список, в котором
„левая“ часть будет меньше $A[i]$, и правая - больше. Аккуратно

все это-быстро. Процедура по принципу рекурсии получим
отсорт. в порядке возраст-а списка.

2) Для решения мы бы могли использовать список, как в задаче 6, но
нам нужно отсортировать его и поэтому используем массив.

3) Создадим массив с элементами S , добавим в него
массивы с результатами вида $B = [(res-m), (uznac-e), (N)]$, пред-но проверив
его значение: $(N \geq 1, N \in \mathbb{N}, X$ - число тало, Y - строка), иначе выведем сообщ. об ошибке

4) Сортируем массив с элементами, используя как функ. утил.
сравнения (в алгоритме они обозначены $A[i], A[j], A[g]$) значениями



рез-тата.

5) После сортировки выведем по порядку все элементы S .



Выходной лист № 2

1) Проверим, что слово $\overset{\wedge}{\text{натуральное}}$, иначе выводим «не-натуральное» и выведем «не-натуральное»



2) Объявим массивы:

$$A = [I, V, X, L, C, D, M] - \text{выход римских цифр}$$

$$B = [1, 5, 10, 50, 100, 500, 1000] - \text{значения}$$

$$C = [a_1, a_2, a_3, \dots, a_n], \text{ где } a_i = (n \% (10^{**i})) // 10^{(i-1)}$$

(например для $n=28 \quad C = [8, 1, 2, 0, 0, \dots]$)

3) Для всех i от 1 до 5, шагом 2:

если $A[C[i+1]] \geq 1$, то $A[C[i]] - 5$ - и получить это

цифры мы уменьшаем на 5 единиц, чтобы и получить (цифры массива «пять», «10», «пятидесять», «пятьсот».

4) ~~представим~~ строка S , $S = ""$

5) Циклом от 7 до 1, шаг -1 прибавим к строке S $A[C[i]] \cdot B[i]$

строку S .

учтем циклический вид $4_{10} = IV_{\text{рим}}$, $9_{10} = IX_{\text{рим}}$ и т.д.

Если $C[i] \equiv B[i+1] - 1$, то будем прибавлять S к концу строки

$$S = A[C[i-1]] + A[C[i]]$$

6) Выводим S - представляем в виде римской цифры

Олимпиада школьников «Надежда энергетики»

□

№ группы

0Г 61-88

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7112

ФАМИЛИЯ САЛАЕВА

ИМЯ АНАСТАСИЯ

ОТЧЕСТВО РУСЛАНОВНА

Дата рождения 28.01.1998

Класс: 11

Предмет информатика

Этап: заключительный

Работа выполнена на 3 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады:

Салаева

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№ 1

1. Вводим натуральное число, если это число равно 1, то на экран выводим I
2. Если число равно 2, то на экран выводим II (т.е. добавляем I к I, I приписываем справа) $\text{II} = \text{I} + \text{I}$
3. Если число равно 3, то на экран выводим III (т.е. добавляем справа II к I)
4. Вводим число 5, при этом на экран выводим V
5. При вводе числа 4 отнимаем от V I (т.е. слева от V подписываем I)
6. При вводе чисел 6, 7, 8 добавляем к V справа I, II, III (т.е. вводим число V, но справа от него добавляем I - если число 6, II - если число 7, III - если число 8)
7. Вводим число 10, при этом на экране выводим X
8. При вводе чисел 9, 11, 12, 13, 14, 15, 16, 17, 18 от числа 10 либо отнимаем либо добавляем такое число в римском виде, чтобы получить введенное число.
9. Вводим число 50, на экране выводится L
10. Вводим число 100, на экране выводится C
11. Вводим число 500, на экране выводится D
12. Вводим число 1000, на экране выводится M.
13. При введении любого число добавляем или отнимаем от известного нам число в римской записи столько, чтобы получилось введенное число. При этом отнимать и добавлять можно только те число, запись которых нам известна в римском виде.



№ 2

Кодирование информации в двоично-десятичной системе счисления целесообразно можно использовать для кодирования очень важной информации, о которой знает определенный круг лиц. Использование двоично-десятичной системы счисления снижает вероятность утечки информации.

№ 3.

1. Вводим модуль n
2. Вводим число a
3. Если модуль n и число a можно разделить без остатка на число большее 1, то мультипликативно обратного числа в нет. В таком случае на экран выводим «-».
4. Если модуль n и число a можно разделить только на 1, то выводим число b .
5. Для нахождения числа b из значения модуля n вводим значение числа a .
6. Полученной результат выводим на экран.

№ 4

Поскольку квадраты расположены в виде спирали, то сделав небольшой чертёж, можно заметить определённый порядок чередования рек. Максимальное число квадратов 15000

2	2	2	2
2	2	2	2
2	2	2	2
2	2	2	2



из-за чередования количество рек можно найти следующим образом:

$$4 + 2 \cdot 3 + 2 + 3 + 2 + 3 \cdot 2 \cdot 2 + 3 + 2 \cdot 2 + 3 + 2 \cdot 3 + 3 + 2 \cdot 3 + 3 + 2 \cdot 4 + 3 \dots$$

$$\dots + 2 \cdot 125 + 3 + 2 \cdot 123.$$

Поскольку число квадратов не должно превышать 15000.

Число рек равно:

$$10 + 3 \cdot 249 + 2 \cdot 110 + 2 \cdot 310 + 2 \cdot 510 + 2 \cdot 710 + 2 \cdot 910 + 2 \cdot 1110 +$$

$$+ 2 \cdot 1310 + 2 \cdot 1510 + 2 \cdot 1710 + 2 \cdot 1910 + 2 \cdot 2110 + 2 \cdot 2310 +$$

$$+ 2 \cdot 2510 + 2 \cdot 242 + 2 \cdot 244 + 2 \cdot 246 + 2 \cdot 248 + 2 \cdot 125 + 2 \cdot 123 =$$

$$= 33653$$

Ответ: минимальное число рек - 33653.

v6

1. Вводим имена игроков Петр и Данил
2. Пусть результат Петра фиксируется в «результат 1», а результат Даниила в «результат 2»
3. Просчитать результат Петра и Даниила на всей странице в каждой клетке
4. Сравнить полученные данные
5. Если число очков у игрока, чьи результат ^{больше} фиксируется в «результат 1», то на экран выводится имя Петра
6. Если число очков у игрока, чьи результат фиксируется в «результат 2» больше, то на экран выводится имя Даниила

Олимпиада школьников «Надежда энергетики»

№ группы

№ группы

Вариант № 4112

шифр

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ГЕЛИВЕРСТОВ

ИМЯ Илья

ОТЧЕСТВО АМТРИЕВИЧ

Дата рождения 02.02.1998

Класс: 11


Предмет информатика

Этап: заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 24.02.15
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



N4

Всегда год. мот не
был

```
program my4;
var N, r, k, i : integer;
    sum: longint;
begin
  readln(N);
  if N=1 then writeln(4);
  if N=2 then writeln(4);
  if N>2 then begin
    sum := 4;
    r := 1;
    while N>2 do begin
      for i:=1 to 2 do begin
        sum := sum + 3;
        N := N - 1;
        k := 0;
if n=2 then break;
        while (N>2) and (k<r) do begin
          sum := sum + 2;
          k := k + 1;
          N := N - 1;
        end;
        if N=2 then break;
      end;
      r := r + 1;
    end;
writeln writeln(sum);
  end;
end.
```





№3

```
program my3;  
var a, b, n, a1, n1, f: integer  
  
begin  
  readln(a, n);  
  if a > n then writeln('Неправильные данные')  
  else begin  
    a1 := a;  
    n1 := n;  
    f := 0;  
    while n1 <> a1 do  
      if n1 > a1 then n1 := n1 - a1  
      else a1 := a1 - n1;  
    if a1 = 1 then  
      for b := 0 to n do begin  
        if (a * b) mod n = 1 then begin  
          f := 1;  
          writeln(b); break; end;  
        end;  
      else writeln('Не существует');  
    if f = 0 then writeln('Не существует');  
  end;  
end.
```





№6

```

program my6;
const
var M, N, i, j, sum1, sum2
    a: array[1..100; 1..3] of integer;

```

Кол-во партий равно:

(кол-во строк - 1)

Перп - 1 строк (sum1)

Данши - 2 строк (sum2)

```

begin
  writeln('Введите количество строк и столбцов клеток');
  readln(M, N);

```

```

  j := 2; sum1 := 0;
  for i := 2 to M do

```

```

    for j sum1 := sum1 + a[i, j];

```

Считаю, что все три столбца таблицы увеличатся в кол-ве клеток

```

  j := 3;

```

```

  sum2 := 0;

```

```

  for i := 2 to M do

```

```

    sum2 := sum2 + a[i, j];

```



```

  if sum1 > sum2 then writeln('Перп')

```

```

  else if sum2 > sum1 then writeln('Данши')
  else writeln('Гурса');

```

```

end.

```

№2

Данная система счисления ^{может использоваться} применяется при работе с дробными числами, с числами с плавающей запятой, а также ~~в перп~~ при записи больших чисел для удобства перевода в другие системы счисления. ⊖

Так же ~~еще~~ применение возможно, если нужно закодировать другие символы помимо цифр, т.к. в ^{код} код не заметны.



N1

you must not

```
program my1;  
var a:array[1..13] of char;
```

```
begin
```

```
readln(n);
```

```
a[1]:= 'I';
```

```
a[2]:= 'V';
```

```
a[3]:= 'V';
```

```
a[4]:= 'X';
```

```
a[5]:= 'X';
```

```
a[6]:= 'XL';
```

```
a[7]:= 'L';
```

```
a[8]:= 'XC';
```

```
a[9]:= 'C';
```

```
a[10]:= 'CD';
```

```
a[11]:= 'D';
```

```
a[12]:= 'CM';
```

```
a[13]:= 'M';
```

```
while n
```

```
if (n mod 10) < 3 then begin
```

```
  x := n mod 10;
```

```
  while x > 0 do
```

```
    s := s + a[i]; insert(s, 1, a[1])
```

```
  end
```

```
else if (n mod 10) = 4 then s := s + a[2]; insert(s, 1, a[2]);
```

```
else if (n mod 10) = 5 then insert(s, 1, a[3]);
```

Олимпиада школьников «Надежда энергетики»

Blank box for group number

№ группы

Вариант № 7102

AN 79-36

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ СЕМЕНОВ

ИМЯ АНДРЕЙ

ОТЧЕСТВО ЮРЬЕВИЧ

Дата рождения 27.10.1998

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

Дата выполнения работы: 27.2.95
(число, месяц, год)

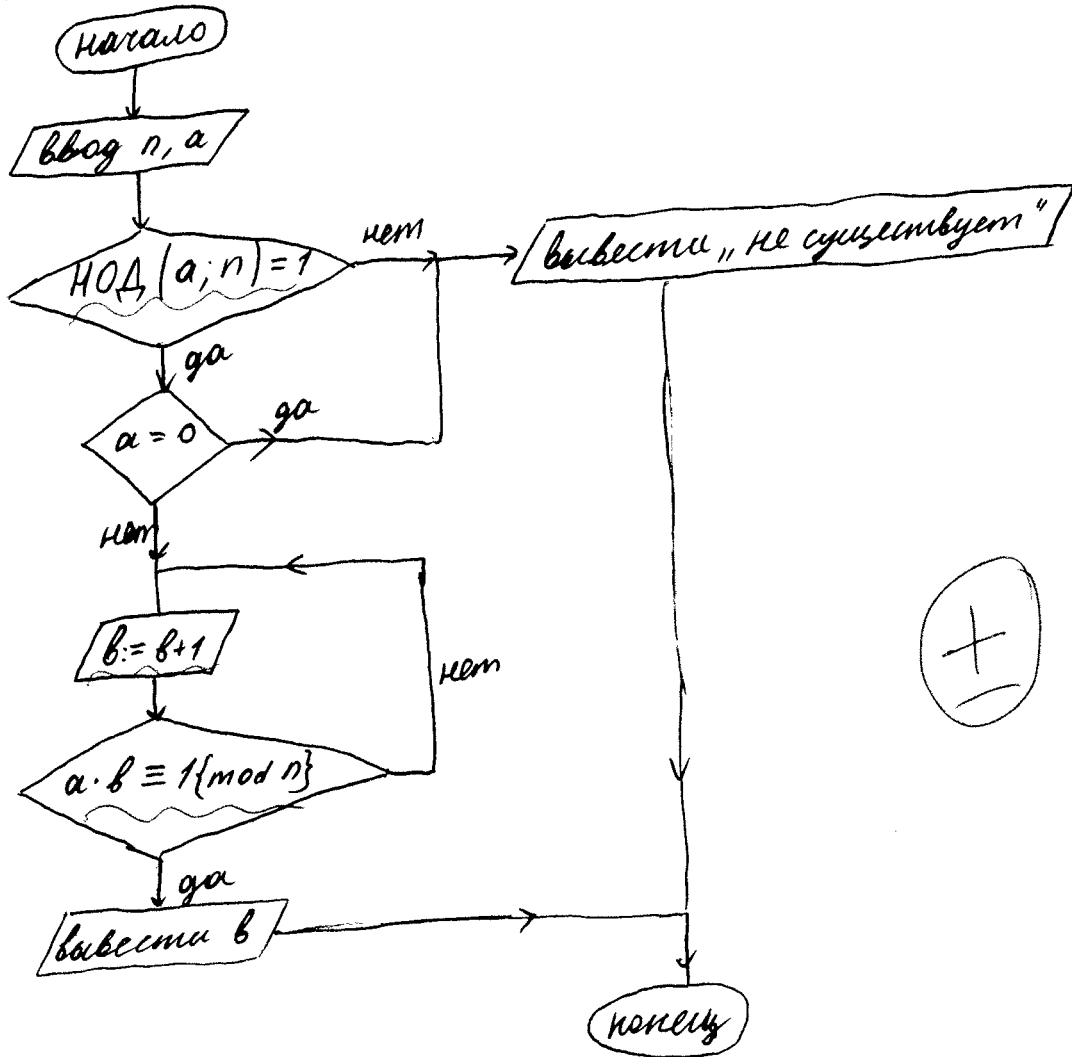
Подпись участника олимпиады: Евд

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.

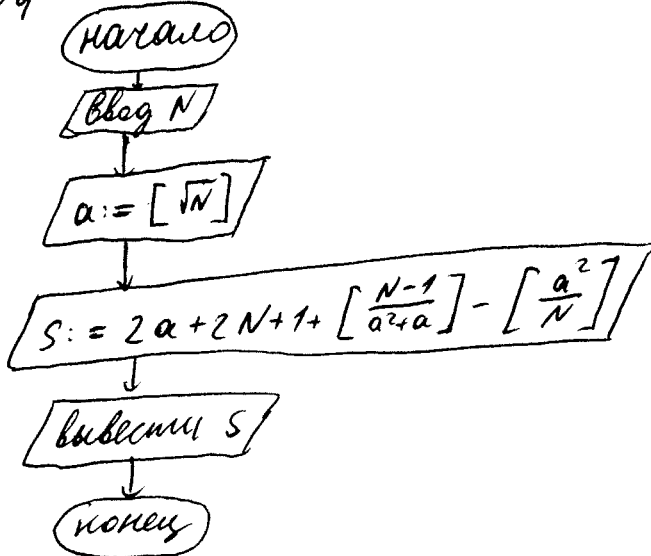


7102

N3



N4





N1

- 1) Сначала ~~представим~~ ^{переведём} число n_8 в десятичную систему счисления:

$$a = ((n \operatorname{div} 10^{l-1}) \bmod 10) \cdot 8^{l-1} + ((n \operatorname{div} 10^{l-2}) \bmod 10) \cdot 8^{l-2} + \dots + ((n \operatorname{div} 10^0) \bmod 10) \cdot 8^0, \text{ где } l - \text{кол-во цифр в числе } n.$$

- 2) Так как римские вместе с цифрами использовали латинские буквы соответствующие значения

10^i или $5 \cdot 10^j$, где $i \in \mathbb{Z}; j \in \mathbb{Z}; i \geq 0; j \geq 0$, ~~то~~ но

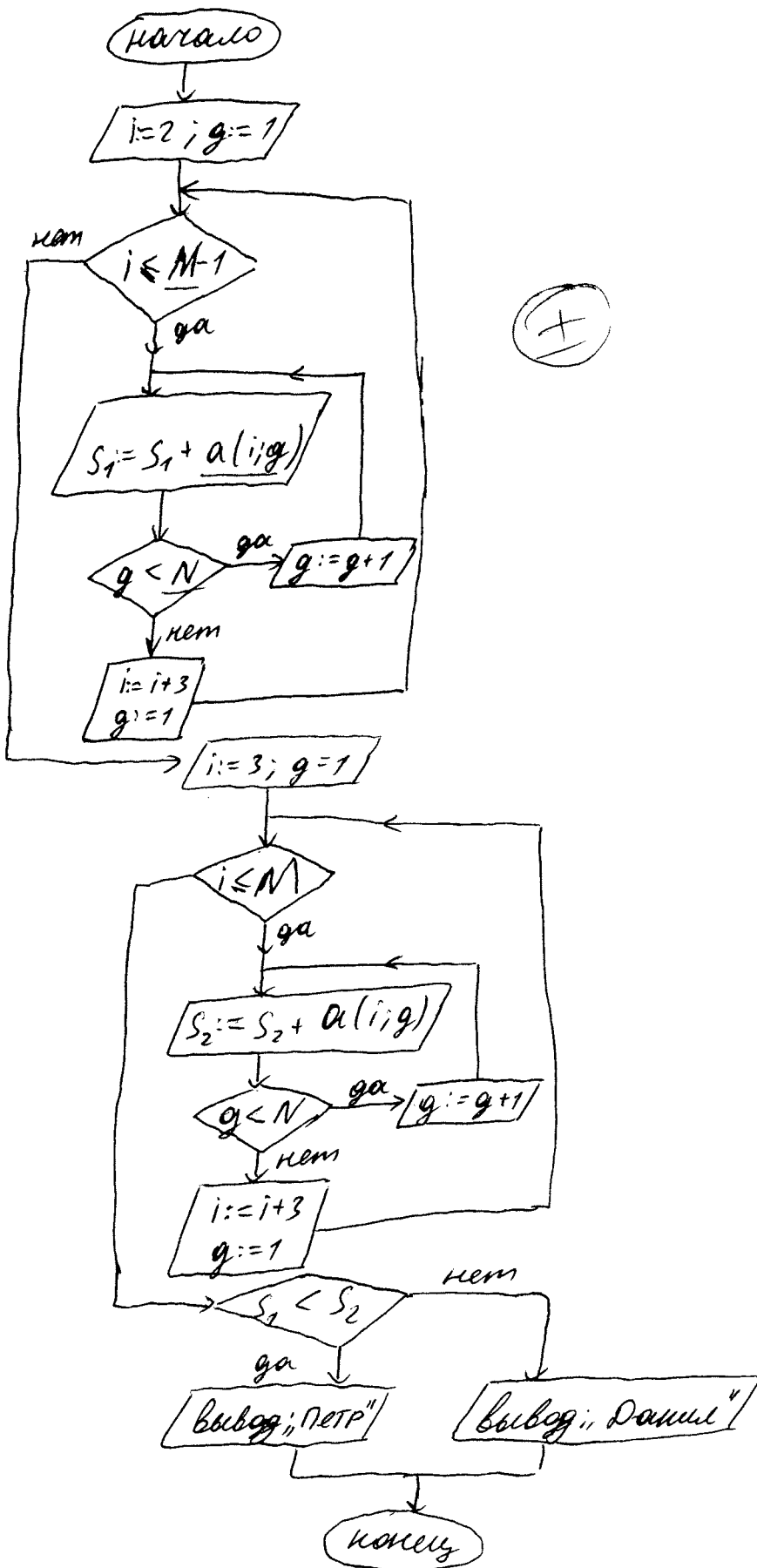
пусть $\{a\}$ - римская буква соответствующая числу вида 10^i или $5 \cdot 10^j$

$$\begin{aligned} \text{Тогда } 2 \cdot 10^i &= \{10^i\} \{10^i\} \\ 3 \cdot 10^i &= \{10^i\} \{10^i\} \{10^i\} \\ 4 \cdot 10^i &= \{10^i\} \{5 \cdot 10^i\} \\ 5 \cdot 10^i &= \{5 \cdot 10^i\} \\ 6 \cdot 10^i &= \{5 \cdot 10^i\} \{10^i\} \\ 7 \cdot 10^i &= \{5 \cdot 10^i\} \{10^i\} \{10^i\} \\ 8 \cdot 10^i &= \{5 \cdot 10^i\} \{10^i\} \{10^i\} \{10^i\} \\ 9 \cdot 10^i &= \{10^i\} \{10^{i+1}\} \end{aligned}$$

- 3) Представив число a в форме $a = a_1 \cdot 10^{k-1} + a_2 \cdot 10^{k-2} + \dots + a_{k-1} \cdot 10 + a_k$, где a_1, a_2, \dots, a_k - цифры числа a , k - кол-во цифр a , и заменив $a_1 \cdot 10^{k-1}, a_2 \cdot 10^{k-2}, \dots, a_k$ на соответствующие составленные римские буквы мы получим ряд латинских букв, соответствующий числу n_8 представленного в виде римских цифр



N6



Олимпиада школьников «Надежда энергетики»

№ группы

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7113

шифр

ФАМИЛИЯ СЕНДЕР

ИМЯ АНАСТАСИЯ

ОТЧЕСТВО ЛЕОНИДОВНА

Дата рождения 31.10.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 5 листах

Дата выполнения работы: 15.03.15
(число, месяц, год)

Подпись участника олимпиады: _____

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



N1.

program z1;

var N, f1, f2, x, y, c1, c2, t, d, i, m : integer; code : integer;
s : string;

begin;

readln(N); {flag десятичного числа}

f1:=1; f2:=1; {два первых числа Фибоначчи}

x:=0; y:=0; {вспомогательные переменные}

s:=''; {строковая переменная для числа в десятичной системе с.с.}

while N > 0 do begin

f1:=1; f2:=1; x:=0; y:=0; {записываем нули для каждого прохода через цикл}

while f1 < N do begin

x:=f1;
y:=f2;
f1:=f1+f2;
f2:=x;

end;

f1:=f1-y;

{алгоритм поиска максимального числа Фибоначчи, которое можно вычесть из данного числа}

N:=N-f1; {разложение данного числа на разряды}

~~c1:=0; c2:=0; t:=0;~~
t:=c1;
c1:=f1;
c2:=t;
d:=0; {c1 и c2 - два числа Фибоначчи, вводимые в разряды; t - вспомогательная переменная}

if (c1 <= 0) and (c2 <= 0) then begin

while x < c2 do begin

m:=x;
x:=x+y;
d:=d+1;
y:=m;

end;

d:=d-1; {алгоритм поиска количества чисел Фибоначчи стоящих между двумя, которое есть в разном, т.е. кол-во нулей}

N2 - нет



```

for i:=1 to d do
  s:=s+'0'; {функция числа нулей}
s:=s+'1'; {функция, прибавляемая, т.к. каждый раз одно число}
end;
end;
t:=0; {вспомогательная переменная используется, чтобы
перевести полученное в итоге число из
строки в число}
val(t,s,code); {перевод из строки s в число t}
write(t); {вывод}
end.

```

NB.

Целеуказание для написания программы и т.д. обратные использовать динамические переменные — списки. Это позволяет экономить ~~память~~ использовать память и оптимизировать обработку полученной информации.

Алгоритм:

объявляем целочисленные переменные M, N, K, X, L, P ;

начало программы

создаем первый список элементов (кол-во $N-M+1$);

каждый элемент списка содержит следующую информацию: число ярусов веток (первый — M , последний — N), количество елей e соответствующим числом ярусов, указатель на следующий элемент списка;

для всех элементов списка

начало цикла

количество елей e и определенное число ярусов присоединить значение $коль$;

конец цикла;



создаем второй список (кол-во элементов $K \cdot 1000 / 2$ -
 K - кол-во километров, перевели в метры и разделили
 на 2, т.к. ель занимает по площади 2 м^2 площадь)

для всех элементов списка (кол-во рядов ели)
 начало цикла

для всех создаем новый список элементов
 (кол-во элементов $L \cdot 1000 / 2$, это сами ели);

каждый элемент списка содержит пару
 число и сортировку: указатель на соответствующий
 элемент списка, число рядов елей веток
 конкретной ели;

конец цикла;

параллельно проходим оба созданных
 списка;

для всех ^(списков) элементов второго списка
 начало цикла

для всех элементов каждого элемента второго
 списка
 начало цикла

вспомогательной переменной x присваиваем

значение, являющееся числом рядов веток ели;

для всех элементов первого списка
 начало цикла
 если x равно числу рядов веток, то

количество елей увеличивается на 1;

конец цикла;

конец цикла;

конец цикла;

для всех элементов первого списка
 начало цикла

выводим сортировку \uparrow элемент:

число рядов веток и количество елей

конец цикла;

удаляем второй список;

конец программы



```

n3
program z3;
var p, a, d, t, r, s, k, i, l, x, N, f, j;
begin
  readln(p);
  x := p - 1;
  s := 0;
  f := 0;
  while f = 0 do begin
    if x mod 2 = 0 then begin
      s := s + 1;
      x := x div 2;
    end;
    if x mod 2 <> 0 then begin
      f := 1;
      d := x;
    end;
  end;
  t := lg(p) div lg(2); { t = log2(p) }
  f := 0;
  for i := 1 to t do begin
    randomize;
    a := random(p - 1);
    N := 1;
    for j := 1 to d do N := N * a;
    if N mod p = 1 do f := 1; { проверка первого условия }
    for j := 0 to s - 1 do begin
      r := j, k := 1;
      for l := 1 to r do k := k * 2;
      N := 1;
      for l := 1 to k * d do N := a * N;
      if N mod p = -1 do f := 1; { проверка второго условия }
    end;
  end;
  if f = 1 then write('первое условие') else write('второе');
end.

```



п 5.

Начало программы
создаем список элементов (т.е. массив, который написан
на листе блокнота (matrix));
инициализируем список перем.
Получим первое число, присваиваем его первому элементу списка;
Получим следующие числа пока не закончится запись Matrix,
для всех элементов списка
начало цикла
проверяем весь список, сравниваем значение текущего
элемента с значениями элементов списка
если оно наибольшее, то вставляем в начало списка
новый элемент, если оно наименьшее, вставляем в конец
списка новый элемент. если ни то, ни другое условие
не выполняется, ищем ему подходящее место
в списке (убывающий порядок) и вставляем
внутри списка;
Получим следующие значения Matrix пока
конец цикла; Получим переменные N и M;
для всех элементов списка
начало цикла
выводим в одну строку M элементов; (⊕)
переходим в следующей строке;
конец цикла;
удаляем список;
конец программы.

п 4

```
program z4;  
var k, i, m, n: int64;  
begin  
  readln(N, M);  
  k := 1;  
  for i := 1 to M do begin  
    k := k * N;  
  end;  
  write(k);  
end.
```

⊖

Олимпиада школьников «Надежда энергетики»

Blank box for group number

№ группы

ИН 28-74

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7103

ФАМИЛИЯ Смаляков

ИМЯ Даниил

ОТЧЕСТВО Сергеевич

Дата рождения 09.09.1998

Класс: 10

Предмет Информатика

Этап: Заключительный

Работа выполнена на 9 листах

Дата выполнения работы: 15.03.15
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



алг Фибс (цел N)

N1

начало

цел: $numb \in N$

$numb[0] = 1;$

$numb[1] = 2;$

цел $pointer = 1;$ $pointer \in N$

пока $numb[pointer] < N$ делаем введем число Фиб по N

кц

$pointer = pointer + 1;$

$numb[pointer] = numb[pointer-1] + numb[pointer-2];$

кц

$pointer = pointer - 1;$

для $i = pointer$ выш до 0 делаем

кц

если $N = numb[pointer] \geq 0$ тогда

~~$N = N - numb[pointer];$~~

~~выведем "1";~~

~~конец~~

иначе, выведем "0";

~~$pointer = pointer + 1;$~~

конец кц

N2

алг Промехитон (цел N1)

начало цел $numb = 0, count = 0$

цел ~~min~~ $min = N_1 + 51;$

для $n = N_1$ до $N_1 + 50$ делаем

кц

для $i = N_1$ до корень из R делаем

кц

для $j = N_1$ до корень из R делаем

кц

для $k = N_1$ до корень из R делаем

кц

для $l = N_1$ до корень из R делаем

кц

если $i \cdot i + j \cdot j + k \cdot k + l \cdot l = R$ тогда

$count = count + 1$

кц

предположим, что корень из R округлен ил до целого числа



N2 (продолжение)

```

кч
кч
кч
если count < min тогда
    min = count;
    minv = R;

```

(±)

```

кч
вывести R minv
конеч.

```

N3

алг Миллер-Рабина (дея IV)

начало

```

цел t, s, d, a, сорч, flag;

```

```

t = log2(p);

```

```

s = 0;

```

```

сорч = p-1;

```

```

пока сорч mod 2 == 0 делать

```

```

кч

```

```

    сорч = сорч / 2;

```

```

    s = s + 1;

```

```

кч

```

```

d = сорч;

```

```

flag = 0;

```

```

если (a в степени d, mod p == 1) тогда

```

```

    flag = 1;

```

```

для i = 1 до t делать
кч.

```



N3 (продолжение)

a = случайное от 1 до p-1

flag = 0;

если (a в степени 2) mod p == 1 тогда

flag = 1;

перейти на следующий шаг

иначе

для j = 0 до S-1 делаем

кф

если a в степени ((2 в степени j) * 2) mod p == -1 тогда

flag = 1

перейти к следующему шагу

кф

если flag == 0 тогда вывести „составное“

кф

если flag == 1 тогда вывести „вероятно простое“

конечу

функция Null принимающая массив целых
чисел и возвращающая логическую переменную
нуль
или
ошибку

функция Null (цел a [])

начало

лог flag = 1;

для i = 0 до последнего элемента массива

кф если a [i] != 0 тогда

flag = 0;

вызову кф функции

~~если a [i] < 0 тогда~~

кф

если flag == 1 вернуть 1, иначе 0



к4 (прочитайте)

конец
 // Ядерьк введем функцию возмещения
 // единицы из массива, предполагаем, что передам
 // сам массив, а
 // не его копию.
 функция minus (a [])
 начало

a[0] = a[0] - 1;

для i = 0 до последнего индекса
 массива из

если a[i] == -1 тогда

a[i] = 0

a[i+1] = a[i+1] - 1;

к4

// основное тело

алг Динамическая сортировка ()

начало

где ~~N ∈ 0 ... 1000000000~~; N ∈ 0 ... 1000000000;
~~M ∈ 0 ... 1000000000~~; M ∈ 0 ... 1000000000;
 пром ∈ 0 ... 1000000000; пром ∈ 0 ... 1000000000;
~~res ∈ 0 ... 1000000000~~; res ∈ 0 ... 1000000000;
 // Max - наибольшее число



вводим поочередно в массив N, и каждое
 число N, так, чтобы первой элементом
 массива N являлся крайний правый цифрой,
 второй являлся второй с правого края цифрой.
 Так же вводим массив M.
 Одинаковые цифры элементов обоих массивов
 обозначим как 0.
 Обозначаем пром, пром, как N.

Пусть t = 1000000



```

пока Null (M) == 0 делаем
нч для
for i = 0 до N-1 делаем
нч для
for j = 0 до t делаем
нч
res[i+j] = res[i+j] + prom[i] * M[j];
кч
кч
for для i = 0 до t-1 делаем
нч
res[i+1] = res[i+1] + res[i] * 10;
res[i] = res[i] / 10;
кч
prom присвоили значения & res
res обнулили
Minus (M & j);
нч
вывести массив res;
конец.

```

~~Введем функцию проверки на отсортированность~~

~~функция check (a, b, c, d)~~
нч начало int flag = 0;
for i = 0 до N делаем
нч
flag;



N5

Введем логическую функцию, задаваемую
 ил и xor-соединением матрицы



функция
названа

check (a[N][M])

```

ил flag = 1;
for для i = 0 до N-1 делать
  ил для j = 0 до M-1 делать
    ил если a[i][j] > a[i][j+1]
      flag = 0;
      вывоу ил файла;
      вывоу ил файла;
  ил
ил для i = 0 до N-1 делать
  ил если a[i][M] > a[i+1][0] тогда
    flag = 0;
ил
ил если flag == 1 тогда вернуть 1,
иначе 0;

```

помощь,
 для сортировки Матрицы (числ N, M)

числ

Matrix [N][M];

```

для i = 0 до N-1 делать
  ил для j = 0 до M-1 делать
    ил ввоу Matrix [i][j];
  ил
ил
ил

```



~~пока check (Matrix[N][M] == 0) делать
 иу ~~до~~~~

~~для i = 0 до N-1 делать~~

~~иу~~

~~для j = 0 до~~

пока check (Matrix[N][M]) == 0 делать

иу

для k = 0 до N-1 делать

иу

для i = 0 до M-2 делать

иу

для j = 0 до M-1 делать

иу

если Matrix[k][i] < Matrix[k][j]

иу поменять местами Matrix[k][i] и Matrix[k][j];

иу

иу

для i = 0 до ^NM-2 делать

~~для~~

~~иу~~

если

Matrix[i][M] < Matrix[i+1][0]

иу

поменять местами

Matrix[i][M] и Matrix[i+1][0];

иу

иу

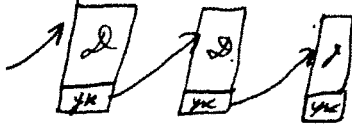
вывести Matrix;

иу



№6

Воспользуемся двусвязным списком, т.к. мы знаем точнее кол-во элементов



Для него определены следующие команды:

- создание нового числа
- добавление элемента в конец
- удаление указанного элемента в начало

Придется создать список списком, чтобы не пропускать указаний

для арг. $Print$ (числ $N, M; L, K, O$)

числ $count \in [M-N]$; $flag = 0$

заполним этот массив 0;

создаем список рывков

~~создаем список и т.д.~~

~~пока на поле есть элемент~~

~~пока все поле под есть элемент~~

~~и~~

~~пока на линии под есть элемент~~

~~и $flag = 0$~~

~~если $flag$ равно $M-N$~~

~~для $i = 1$ до N увеличь~~

~~и~~

~~если указание под заведи~~

~~счит i и т.д. для~~

~~если указание заведи $N-1+i$ т.е.~~

~~пока $count \in [M-N]$;~~

~~$flag = 0$~~

~~обозначить элемент списка т.е. т.е.~~



добавим элемент ^{в начало массива} в конец списка u
 1) \rightarrow если $flag = 0$, тогда ~~вывести~~
 ku

ku
 ku
 вывести массив символов $N-N3$;
 конец.

В начале стоит добавить проверку
 если $N \geq 0$ или $M < 0$ или $L < 0$ или $K < 0$

или $Q < 0$ тогда вывести "Неверно вве-
 дены данные"

Также массив символов

1) Если $flag = 0$ (значит нет ~~элементов~~
 ни элементов, тогда
 добавим элемент в список -1
 добавим новый элемент в список
 u списка u по условию задачи.

Так в массиве сформируется
 список u по условию задачи,
 а в списке u элементов u
 еще массив u по условию задачи.

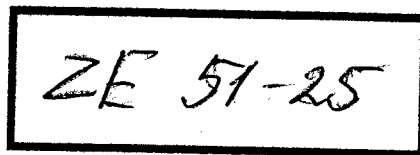
выдан драгматический лист

Олимпиада школьников «Надежда энергетики»



№ группы

Вариант № 7092



шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Соловьев СОЛОВЬЕВ
ИМЯ Ирмий СЕРГЕЙ
ОТЧЕСТВО Андреевич АНДРЕЕВИЧ

Дата рождения 20.09.1999 г

Класс: 9

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады: _____


Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



n 5

Программа на языке PASCAL:


```
VAR
N: REAL;
BEGIN
N := 10000;
WHILE (n mod 7 = 0) and (
WHILE (N >= 10000) and (N < 100000000) do
  BEGIN
  IF (N MOD 7 = 0) and (N MOD 11 = 0) THEN WRITELN(N);
  N := N + 1;
  END;
END.
```



n 4

Программа на языке PASCAL:

```
VAR N, M: longint;
I: INTEGER;
BEGIN
KREAD (N, M);
FOR I := 2 TO M DO
  BEGIN
  N := N * I;
  VAR N1, M1, M: longint;
  I: INTEGER;
  BEGIN
  READ (N, M);
  N1 := N;
  FOR I := 1 TO M DO
    BEGIN
    N1 := N1 * I;
    END;
  WRITELN (N1);
  END.
```





и 1

Позиционная система счисления — система счисления, в которой «цифры» той или иной цифры числа зависят от её позиции/значности и требуемому разряду: тем ближе к цифра к порядку и требуемому разряду, тем меньше её «цифры», и наоборот, тем дальше цифра от нулевого разряда, тем больше её «цифры».

Рассмотрим две десятичные шестнадцатеричные числа 291_F и цифра «9» имеет разряд десятков, и её роль гораздо больше, чем роль той же цифры «1», которая расположена в разряде единиц.

Также возведем число с факториальными цифрами для ~~два~~ большей полноты.

$$3333_F = 3 \cdot 4! + 3 \cdot 3! + 3 \cdot 2! + 3 \cdot 1! = 72 + 18 + 6 + 3 = 99_{10}$$

Таким образом, «цифры» цифры (в нашем случае, «3», отрицательные её показатели в числе «3», показатели в разряде тысяч, имеют гораздо большее значение, чем «3», показатели в разряде единиц (сравните: $72 \gg 3$) ⇒ система счисления такого типа имеет позиционность.

Ответ: да, является.

и 3

Программа на языке PASCAL:

```
VAR
N, N1, A, K: INTEGER;
BEGIN
READ(N);
N1 := N;
K := 0;
WHILE (N1 < 2 * N) DO
```

(см. на сл. стр.)



```

BEGIN
A := N1 mod N;
IF (N DIV A <> 0) THEN K := K + 1;
N1 := N1 + 1;
END;
WRITELN (K);
END.

```

~ 2

Туроправа на языке PASCAL:

VAR

i, GT, T : INTEGER;

~~G: ARRAY [1..27] OF~~

G: ARRAY [1..27] OF INTEGER;

G[1] := -12

G[10] := -3

G[2] := -11

G[11] := -2

G[3] := -10

G[12] := -1

G[4] := -9

G[13] := 0

G[5] := -8

G[14] := 1

G[6] := -7

G[15] := 2

G[7] := -6

G[16] := 3

G[8] := -5

G[17] := 4

G[9] := -4

G[18] := 5

BEGIN

WRITE ('Наименование тура: ');

READ (i);

WRITE ('Введите туровое время: ');

READ (GT);

T := GT + G[i];

WRITELN (T);

~ 4

VAR

K, A, n, i: real;

b: array [1..1000] of real;

+

(ил. см. стр. 10)



```
begin
read(n); n := 1;
for i := 1 to 44 do
  begin
    n := n * i;
  end;
a := length(n);
b[1] := 10;
for k := 2 to a do
  begin
    b[k] := b[k-1] * 10;
  end;
k := a;
while (k > 0) do
  begin
    if (n div b[k] = 4) then write(k, ' '); // " " - пробел //
    n := n mod b[k];
    k := k - 1;
  end;
end.
```

Олимпиада школьников «Надежда энергетики»

Blank box for group number

№ группы

Вариант №

7102

Box containing the code CW59-89

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ

Сорокин

ИМЯ

Иван

ОТЧЕСТВО

Исупрович

Дата рождения

21.09.1998

Класс:

10

Предмет

ИНФОРМАТИКА

Этап:

Заключительный

Работа выполнена на 3 листах

Дата выполнения работы:

27.02.15
(число, месяц, год)

Подпись участника олимпиады:

Handwritten signature

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№6.

И.А.

ввод M ;ввод N ;

перемешанное

petr ≥ 0 ; // сумма результатов Петра.dan ≥ 0 ; // сумма результатов Даниила.

h; // высота таблицы результатов

l; // ширина таблицы результатов.

i; // счёт элементов массива

если $M > N$, то $M = h = M$, иначе $h = N$; // узнаем как вымерит
 $l = N$; // таблица результатов

если $h < 2$ или $l < 3$, то вывод "Неправильные размеры таблицы".
 иначе

представим таблицу результатов в виде двумерного массива $a[h][l]$;

$i = 2$ до h

н.ц.

petr = petr + $a[i][2]$;dan = dan + $a[i][3]$;

к.ц.

если petr = dan, то вывод "Ничья";


если petr > dan, то вывод "Дания";

если petr < dan, то вывод "Пётр";

К.А.

№2.

Это будет целесообразно, если нам нужно или выгодно
 работать с бинарными цифрами этого месяца.

Проект в двоичной системе и так будет выигрывать много,
 мы будем работать именно с бинарными цифрами месяца и
 сможем преобразовывать только месяц; 

А в двоично-десятичной код мы сможем преобразовывать сами цифры
 месяца, менять их местами, записывать и т.д.

№3 - тест



N4.

Н.К.

выбор N;

переменные

S=10; // сумма рек

Sch=4; // остаток квадратов

p=1; // кол-во рек между тремя.

t, i; // для повтора

Если $N < 1$ или $N > 15000$, то выбор "Неверное значение N".

иначе

если $N = 1$, то выбор "4",

иначе

если $N = 2$, то выбор "7",

иначе

если $N = 3$, то выбор "10",

иначе

н.ч. пока sch ≤ n

н.ч.

i = 1 go 2

н.ч.

t = 1 go p

н.ч.

S = S + 2;

Sch = Sch + 1;

если sch = n + 1, то остановить все циклы;

к.ч.

S = S + 3;

Sch = sch + 1.

если sch = n + 1, то остановить все циклы;

к.ч.

p = p + 1.

к.ч.

выбор S;

к.ч.

К.А.





N5.

Н.А. Высота таблицы равна максимальному значению $\sqrt[n]{n}$.

$$h = \max \sqrt[n]{n}; t = 2;$$

Возьмем два двумерных массива: $a[h, 3]$ - исходная таблица результатов и $b[h, 3]$ - результирующая таблица в ходе алгоритма.

~~$p = 2$~~ до h пока массив $a[h, 3]$ не будет пустым

н.ч.

$$i = 2 \text{ до } h$$

н.ч.

просмотр всех элементов $a[i, 3]$ и находиме макс

к.ч.

$$i = 2 \text{ до } h.$$

н.ч.

если $a[i, 3] = \max a$

н.ч.

$$b[t, 1] = t; b[t, 2] = a[i, 2]; b[t, 3] = a[i, 3]$$

к.ч.

$$t = t + 1;$$

уменьшить строку равна макс, и сдвинуть массив $a[h, 3]$ ~~с~~ не было пустых строк.

к.ч.

если иначе, то выбор массива $b[h, 3]$

к.ч.

N1.

Н.А.

$$\text{вход } n; p = |n|; s = 0; t = 0;$$

while $p \neq 0$

$$\text{н.ч. } q = p \bmod 10$$

$$s = s + q \cdot 8^t;$$

$$p = p \cdot \text{div } 10;$$

$$t = t + 1;$$

к.ч. если $n > 0$, то $n = p$, если $n < 0$, то $n = -p$.

или перевести число n из десятичной системы в восьмеричную.

Теперь выберем для числа n соответствующее число в римской системе счисления.

к.ч.

Олимпиада школьников «Надежда энергетики»

Blank box for group number

№ группы

JM 25-69

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7112

ФАМИЛИЯ Столная

ИМЯ Анастасия

ОТЧЕСТВО Владимировна

Дата рождения 04.01.98

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 2 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:

[Signature]

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



6.

```

var mas: array [1.. M·N] of integer, n, x, y, z: int 64; a: text;
begin
  assign (a; in.txt)
  reset (a);
  x := 1
  while not eof (a) do begin
    read (a; mas[x]);
    inc (x)
  end;
  x := 2;
  y := 3;
  repeat
  begin
    n := n + mas[x];
    x := x + 3;
    z := z + mas[y];
    y := y + 3
  end;
  until (x > length mas) or (y > length mas)
  if n > z then writeln 'Темп победы';
  if z > n then writeln 'Давнее победы';
  if z = n then writeln 'Ничья';
end.

```



5.

Хранилище - таблица, где в строке x : 1 - N^2 ; 2 - значение; 3 - результат.

Алгоритм.

1. Пройти таблицу с начала до конца. Для любых 2-ух соседних строк проверка - если в строке x значения $[x; 3] >$ значения $[x+1; 3]$, тогда:
 2. строка $y =$ строка x ($[x; 1], [x; 2]$) и $[x; 3]$ соответств. присв. к $[y; 1], [y; 2], [y; 3]$)

3. Элементы строки x присваиваются соответв. знач-я x -ой y $[x+1]$.

4. Элементы $[x+1]$ присв. соответв. знач-я из y

5. Были ли произведены перестановки на шагах 1-4? Если да, то выписать 1-4 еще раз, если нет - элементы упорядочены по возраст-ю;

$N=2$ - нет $N=3$ - нет



алгоритм завершён.

2.

Целесообразно применять при записи больших чисел; при необходимости быстрого последующего перевода в 10-ую систему счисления; при необходимости передать последовательность цифр, не являющуюся числом.



4.

Количество рёбер, необходимое для острого квадрата является ~~2~~ числом, а весь числовой ряд имеет формулу $4 + 3 + (3 + 2 \cdot x)$, где x увеличивается на 1 после каждой 2-ух повторений цепочки $3 + 2 \cdot x$

Таким образом, иск. число = $(3 \cdot (N : 2)) + (2 \cdot (N : 2))$



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ СУХОВА

ИМЯ ДАВГА

ОТЧЕСТВО ЦГОРЕВНА

Дата рождения 09.04.1998.

Класс: 11

Предмет информатика

Этап: заключительный (очный)

Работа выполнена на 5 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады: Сухов

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



Вот же
2w

Способ 1: табличный:

Найдем такое число фиббоначи, чтобы оно было больше нашего, но предыдущее (перед ним) было меньше нашего.

51. Пусть оно k , его номер n .

число фиб. k ← L_n с порядк. номер.

Пусть r - изначальное число.

Теперь мы знаем, что длина нашего числа $n-1$. вычитем из r $k \cdot 10^{n-1}$.

Теперь перебираем по порядку числа фиббоначи от $n-2$ до 1. Если наше k \leq число фиб, то пишем 0. Если больше, то пишем 1 и вычитаем из r фиб. число. Так пока ~~ручка~~ r не станет ≤ 1 .

Исть искомое число.

→ 37.

первое фиб. число больше него: 55.

до него идет 34. ⇒ запишем 1

до 34 идет 21. ⇒ пишем 0.

до 21 идет 13. ⇒ пишем 0.

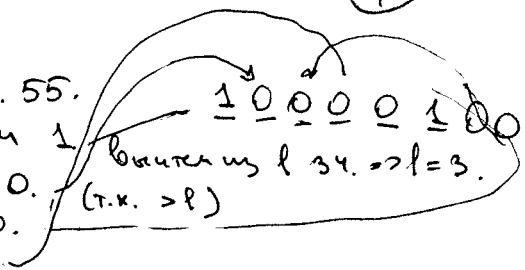
до 13 идет 8. ⇒ пишем 0.

до 8 идет 5. ⇒ пишем 0

до 5 идет 3. ⇒ пишем 1 (т.к. $< r$)

до 3 идет 2 ⇒ пишем 0

до 2 идет 1 ⇒ пишем 0.



вычитаем из r 34. ⇒ $r=3$. (т.к. $> r$)

$l := r - k \cdot 10^i$, где $i=3$ т.к. 3-й порядк номер числа 3.

⊛ Как искать число фиб?

Получим 10000100.

Создаем массив так, что $k^{L_0} = 1$ $k^{L_1} = 1$.

и дальнейшие числа: $k^{L_i} = k^{L_{i-1}} + k^{L_{i-2}}$

пока $k^{L_i} < r$ это повторяем

если $k^{L_i} > r$, то $i=n$. и начинаем раскладывать.

≠



52.

Сделаем «умный» перебор чисел от N_1 до N_1+50 .

$min := -1000; p := 0$

Пусть i - числа от N_1 до N_1+50 . (по очереди)

$p := 0$. (р делает равным нулю)

Пусть j - числа от 1 до $\lfloor \sqrt{i} \rfloor$

Пусть k - числа от 1 до $\lfloor \sqrt{i-j^2} \rfloor$

Пусть l - числа от 1 до $\lfloor \sqrt{i-j^2-k^2} \rfloor$

Пусть s - числа от 1 до $\lfloor \sqrt{i-j^2-k^2-l^2} \rfloor$.

Если $i = j^2 + k^2 + l^2 + s^2$ то:

если $j > k > l > s$ или $j > k > l = s$ или $j > k = l > s$ или $j > k = l = s$ или $j = k > l > s$ или $j = k > l = s$ или $j = k = l > s$ или $j = k = l = s$ тогда

используем

⌊ ⌋ отбор дробкой
част, если
такая есть
т.е. $\lfloor \sqrt{3,2} \rfloor = 3$.

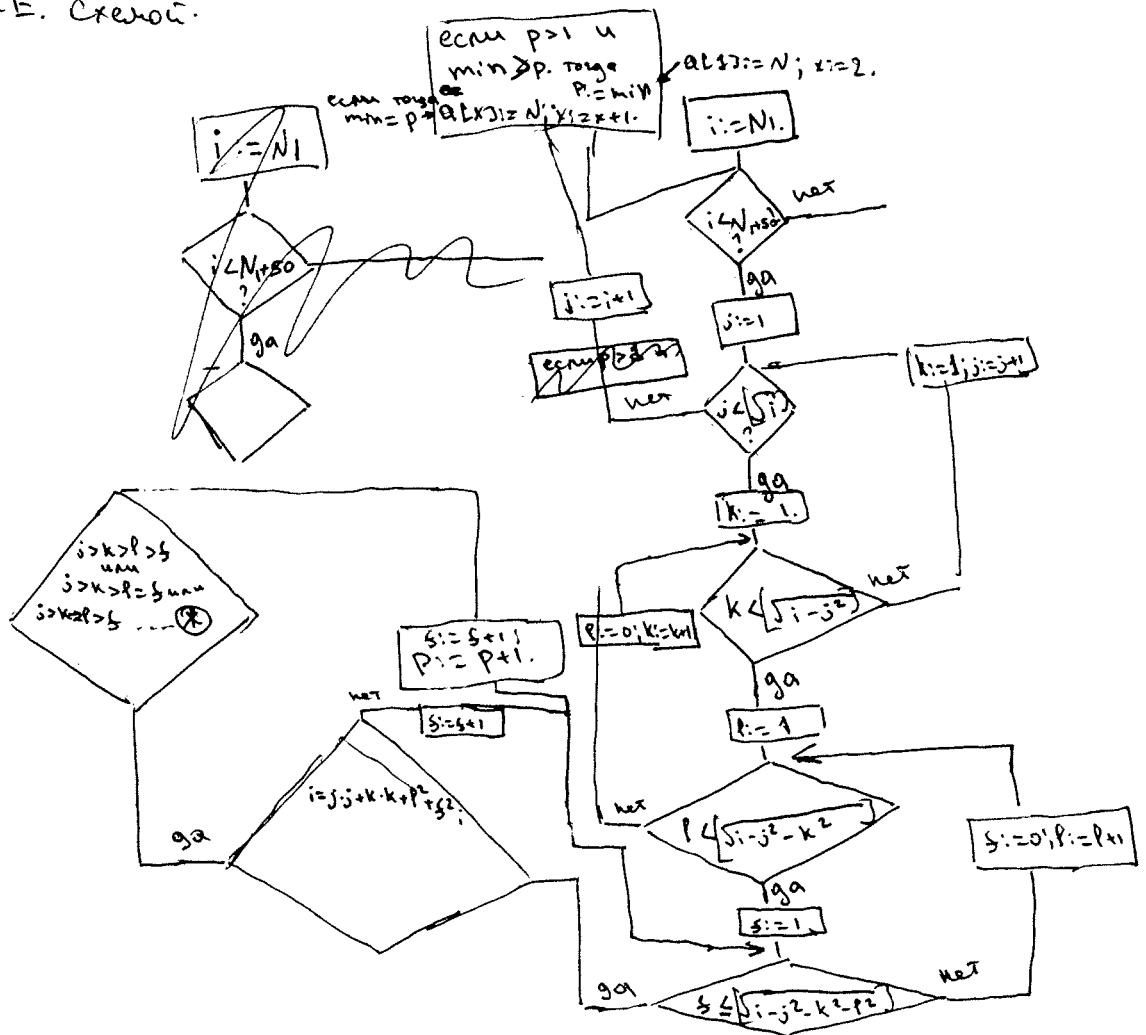
чтобы
случайно
не
учесть
несколько
раз одинако-
вые варианты



прекращаем все переборы кроме i (заканчиваем перебор $jklfs$).

Смотрим, если $p > 1$ и $p \leq min$ тогда $min = p$.

т.е. схемой.





§3:

Перебор sd :{ перебором найдем s и d).1 $s := \text{от } 1 \text{ до } \lfloor \log_2(p-1) \rfloor$; выполнить строку 22 $k := \text{от } 1 \text{ до } \lfloor \frac{p}{2} \rfloor$ выполнить 3 и 4.3 $d := 2k - 1$.4 если $p-1 = 2^s \cdot d$ то закончим перебор. sd найдем.

Наконец проверить:

1 $A := \text{от } 1 \text{ до } p-1$ выполним 2, 3, 4, 52 $bool := false$;3 если $a \bmod p = 1$ тогда $bool := true$;4 $r := \text{от } 0 \text{ до } s-1$ выполним 5.5 если $a^{2^r} \bmod p = p-1$ или $a^{2^r} \bmod p = -1$ тогда $bool := true$.6. если $bool = false$ тогда номер «оставное» и выходим из прог.Если $true$

7. если номер «вероятно простое».



§5.

считываем массив размера $M \times N$. (одномерный!). т.е.

1,1	1,2
2,1	2,2

→ 1,1; 1,2; 2,1; 2,2.

Сортируем пузырьком.

печатаем. от 1 до $2N$

Если надо написать и в строку, то печатаем в 2 строке

от 1 до N , во второй от $N+1$ до $2N$ пока не будет от $(n-1) \cdot N$ до $M \times N$.

```

for i = n to downto 1 do
  for j = 0 to i-1 do
    if a[j] > a[j+1] then swap(a[j], a[j+1])
  
```

т.е. элемент a_n , что сначала n -ое точно станет наименьшим, затем $n-1$ станет меньшим из оставшихся и т.д.



Владимир

ШИФР НЕ ЗАПОЛНЯТЬ! ⇨

5Г 94-29

54.

~~Если дана масса θ , то найти t .~~
(хотела одно)

Занимем число N в массив и перевернем.

т.е. $105 \rightarrow [5|0|1]$ с M -аналогично.

Затем создадим двумерный массив M , в котором $N \times N$ строк.

$M[i] := N$.
 M раз делаем следующее:

1) создаем a такой, что он $M \times N$ - строк
зубируем M , во все строки
в i -ую строку умножаем на i элемент массива N .
(каждый её элемент).

расширяем так, чтобы в i элементе была i цифра.

т.е. ~~18 32 6~~ → ~~21 26~~
18 32 6 → 8 9 9 2

т.к. $18 \bmod 10 = 8$.
 $18 \div 10 = 1 \Rightarrow 32 \neq 1$.
тогда $32 \rightarrow 9$.

$a[i+1] := a[i] \div 10 + a[i+1]$.
 $a[i] := a[i] \bmod 10$.

Внимательно следит, если выхоит за размер - это надо учесть.

далее i элемент сдвигает на i знаков,
2 т.к. на 1
 N т.к. на $N-1$.

получим:

~~221~~ 21 2 12 2 раз

↓
12 → 12 → 24 → 24 → 24
12 → 24 → 24 → 24

далее сложим

a_1, a_2, a_3
 b_1, b_2

a_1+b_1, a_2+b_2, a_3



Складываем все элементы по столбцам. раскладываем так, чтобы в i элементе i цифра. помогло как сделали M раз. печатаем (каждый)



№6.

Собираем данные, сколько елей-какого вида.

Составляем массив из элементов, размера такого, сколько самшитов.

*-Эти элементы из 3х массив. 1-сколько эрмеев веток
2-ее место по ох
3-ее место по оу.

Если какое-то ель умирает или умирает, то
из массива удаляется ~~е~~ и все сдвигается на 1.

(или если хранить с помощью дин. памяти, то
указатель следующего сдвигается на следующий).
Если вырастает новая ветка - меняем по
ее координате индекс эрмеев.



Олимпиада школьников «Надежда энергетики»

№ группы

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7113

шифр

ФАМИЛИЯ Тимофеев

ИМЯ Валерий

ОТЧЕСТВО Витальевич

Дата рождения 23.10.1997

Класс: 11

Предмет информатика

Этап: заключительный

Работа выполнена на 4 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады:

Витальевич

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



① Введи число, например 37. Разложи его на простые
 так, что эти самые простые будут являться числами Фибоначчи.
 Для 37 такими числами будут являться 34 и 3. 34 не
 имеет раскладывать дальше, равно как 3, т.к. две единицы
 рядом стоять не могут. Необходимо раскладывать именно на 34 и 3.
 Количество вариантов множество, условия лишь два: 1- эти
 числа не являются последовательными в ряду Фибоначчи, и 2-
 эти числа принадлежат ряду Фибоначчи. Записав эти
 простые пробежав по циклу, шаговому циклу, задавая
 условие выхода, где ~~число~~ конечное число счетчика
 равно порядковому номеру своего первого простого.
 Например, условием выхода из нашего цикла
 будет являться так: когда счетчик "N" примет
 порядковый номер числа 34, т.е., когда $N=8$. По тому
 мере шага каждого цикла проверим если на этом
 порядковом номере свой численка нашего простого,
 то пишем в строку вывода "1", иначе "0". Например:
 при $N=1$, число Фибоначчи будет 1, ставим "0".
 при $N=3$, число Фибоначчи будет 3, ставим "1"
 и т.д.

Таким образом, в конце мы получим последовательность
 из "1" и "0", что и будет являться числом Фибоначчи.

③ Введи число p.

Представим число $p-1$ в виде $p-1 = 2^s \cdot d$, где

d - нечетное число. Тогда создаем цикл с ~~состоянием~~ ^{предельным}
~~переменной~~ ^{где условие выхода (a=0).}
 $a = 0 \vee 1 \dots (p-1)$, ~~где~~ проверим ~~счетчик~~ ^{на} каждом шаге.



Например: если перемешивание равно 1, то проверяем 1, если 2, то 2; и т.д. до $(p-1)$. Пусть a - перемешивание.

Проверяем по следующим условиям:

Если $a^d \not\equiv 1 \pmod{p^2}$, тогда

Если

создаем цикл ~~с перемешиванием~~, пусть ~~считаем в этом~~ ~~цикле~~ q , тогда цикл ~~с~~ $0 \leq q < p-1$ с предположениями ~~где~~ ~~создаем~~ $(q \neq 0) \neq 0 \pmod{p}$ ($1 \leq q < p-1$), то выполняем цикл в котором каждый раз увеличиваем значение переменной q $0 \leq q < p-1$ до тех пор, пока условие не окажется правильным, а именно:

$$\left(a^{(2^q)} \right) \left(a^{(2^{p-1})} \cdot d \right) \equiv -1 \pmod{p^2}.$$

(На самом деле этим уравнением можно

договориться изначально обозначить условие выхода из цикла, т.е., как только оно становится правильным, мы прививаем выход из цикла) Однако, если уравнение правильное прививаем $q=1$. Выходим из цикла.

Если $q=1$, то просто продолжим перебирать значения ~~значения~~ ~~каждого~~ ~~обернем~~ цикла, т.е. каждого повторения.

Если ~~мы~~ после выполнения цикла основного, q окажется равным 1 то число вероятно простое, иначе число составное.

Конечная проверка: Если $(q=1)$, то выводим вероятно простое, иначе - составное.

Однако, я не учел одну важную деталь.

Вместе с Нилера - Рабина эти проверки выполняются для ϵ случайно выбираем a . Тогда всегда правды.



Проверки внутри цикла обантуются тем же, однако если цикл будет от 1 до t или же другое условие $q=0$, где t - это $\log_2(p)$. число a будем выбирать случайное от 1 до $p-1$.

④ Выпишем N^M , где N и M могут быть от 1000 цифр.

Естественно, что ни один язык программирования не способен обработать столь огромные числа, однако для этого существует длинная арифметика, не позволяющая считать, а позволяющая обрабатывать ~~данные~~ или числа, однако записывая каждый ~~разряд~~ цифра записывающаяся разряд в новом числе. \oplus

⑤ Мы имеем лист, размером M на N клеток. Создаем двумерный массив размером $6M$ строк и M столбцов. Построим пробегаемая по двумерному массиву меняя рядом стоящие числа, элементы массива, взвешивая от их величины, если два соседних не равны, а \ominus последующий больше предыдущего, то меняем их местами, иначе оставляем как есть. Этот способ сортировки называется «пузырек». Самым простым и примитивным,



№5 (продолжение)

Зато действенный способ. Таким образом
максимальный элемент окажется на первом месте
и весь массив будет отсортирован.

6) Создаем двумерный массив K на L ки, тогда
как его размер не более 2 м^2 . Переводим все в
шеры, тогда получаем размерность шага 1м. Следовательно,
проходимся по строкам массива и на каждой
клетке строки проверяем "ель". Входим в цикл от M до N .
Каждый шаг (повторение) цикла проверяем;
Если есть ветка, то инкрементным переменную
счетчик. Как только посчитали кол-во ~~ва~~ уровней веток
инкрементным ~~таб~~ счетчик переменную, которая отвечает
именно за эти с этим кол-вом уровней. Таким образом
"пробегая" по всему массиву-потом. В конце просто
выводим итог. На выход вывели переменные и подружили
состоит уровней.

7) Создаем цикл от $(N+1)$ до $(N+50+1)$ на каждом шаге
проверяем число N (которое изначально равно $N+1$), как до
повторение цикла число N увеличивается на 1 до $N = N+50$.
Раскладываем число N всеми возможными способами на
слагаемые и проверяем являются ли они квадратами, если да, то
инкрементным счетчик a . Выводим на экран, после
выполнения ~~таб~~ счетчик a . Он и покажет кол-во
всех чисел от $(N+1)$ до $(N+50)$.

Олимпиада школьников «Надежда энергетики»

Blank box for group number

№ группы

Вариант № 7103

Box containing the code: 0A 30-49

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Тихонова

ИМЯ Барвара

ОТЧЕСТВО Андреевна

Дата рождения 15.09.1998

Класс: 10

Предмет информатика

Этап: заключительный

Работа выполнена на 5 листах

Дата выполнения работы: 15.03.2015
(число, месяц, год)

Подпись участника олимпиады:



Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№1.

Пусть есть функция $fib(n)$, возвращающая число Фибоначчи по порядковому номеру n . Результат будет храниться в переменной S , где изначально записано то количество нулей, которое будет длиной ~~каждого~~ искомого числа. (В цикле пока $fib(m)$, изначально равный 1 ($m=1$), меньше данного числа X , увеличивает m на 1. Затем (после выхода из цикла) вычитает 1 из m . S содержит m '0'.)

Затем, пока X не равен 0, вновь выполняем цикл (пока $fib(m) / (m=1) < k$, $m += 1$), вычитаем 1 из m .

из k вычитаем $fib(m)$, затем на место m в строке S вставляем '1'.

код на C++:

```
for for (m=1; fib(m) < k; m++);
```

```
S = '0' * m;
```

```
while k != 0 {
```

```
  for (m=1; fib(m) < k; m++);
```

```
  m--;
```

```
  k -= fib(m);
```

```
  S += '1';
```

```
fib(n) {
```

```
  if (n == 1) return 1;
```

```
  else f(1, 2, 2, n);
```

```
}
```

```
f(pre, cur, level, n) {
```

```
  if (level == n)
```

```
    f(cur, cur+pre, level+1, n);
```

```
  else if (level == 0)
```

```
    return (cur);
```

```
}
```



№5.

Сначала нужно ~~не~~ считать все числа и занести в список, а затем отсортировать его по той сортировке n -кратного возмещения.

arr - массив чисел;
 tmp - переменная с символом;

a - список, изначально пустой;

$LIST$ - метод

Представим $LIST$ как двумерный массив символов $[M, N]$.

Тогда в цикле пройдём по всей матрице, добавляя arr в a , если $tmp = ' '$ (пробел) или \neq доминирующая цифра. 10 и приравляя ~~цифру~~ tmp к нулю.

Список a - неотсортированный список всех чисел с $LIST$.

В цикле $i \rightarrow n-1$ пройдём по всем элементам списка, ~~также~~ во вложенном цикле $j \rightarrow n$ пройдём по элементам с $i+1$ до последнего, сравнивая $a[i]$ и $a[j]$. Если второй меньше первого - меняем их местами.

В итоге получаем отсортированный список чисел a с $LIST$.



№4.

Каждым все простое делители числа M , а затем возведем число N в степень каждого простого делителя M . Однако, если число M простое или его простое делители велики, то разложим на простые множители M и перемножим все N в степенях этих простых множителей.

Однако, так как записано только ограничение по размерам N и M , ~~то~~ по ним операций по времени и памяти, то можно просто

взять от 1 до M только те результаты N^k изначально равен 1 по N , но данный способ не самый малый по времени.

Также можно найти ближайшую к числу n степень двойки 2^k и превести n в 2^k с помощью двойки, затем найти разность $n - 2^k$ и вновь найти ст. двойки. Затем

перемножить все N в степенях 2^k , т.к. для нахождения числа в степени 2^k необходимо вычислить n с умножением числа на само себя k раз.



2. Пусть $f(n)$ - функция, возвращающая True, если n можно однозначно представить в виде суммы ~~четырёх~~ ^{двух} квадратов натуральных чисел.

В ней хранится локальный ~~массив~~ ^{адрес} множество всех таких чисел, сами числа находим перебором в циклах (4 вложенных цикла, каждый из которых от 1 до \sqrt{n}), в последнем цикле - проверка найденных чисел на равенство и проверка того, является ли их сумма числом n .)

Проверяет на равенство множеств n и n и возвращает True, если не все они равны между собой.

Затем находим все ~~числа~~ ^{числа} n на заданном ~~масиве~~ ^{диапазоне} и создаем ассоциативный массив с ключом n .

Итерируем по диапазонам от ключа до ключа + 50 ищем числа n и их количество будем записывать от ключа в ассоц. массиве. Выводим те ключи, значение элементов от которых минимально.



№ 6.

Данные описания всего хранения в таблице вида:

№ параметра	1	2	3	4	5	...	Q
	x					...	
	y					...	
	аргусы					...	
	мил?					...	

$\{ \begin{matrix} \text{координаты} \\ \text{координаты} \end{matrix} \}$
 кол-во аргусов $\{0, 1\}$
 есть ли элемент
 сет-овое (0/1)

для построения аргусов создадим список с номерацией от $N_{доМ}$ и составим все элементы матрицы, где δ "аргус" $= 1$ и δ "мил?" $= 1$.

Также для упрощения работы с таблицей можно указать элементы последнего параметра которых $= 0$.

№3 сначала найдем d и S . Их может быть много. Выходим, пока d не станет четным. Мы присваиваем d значение $(p-1) \cdot 2^S$, затем уменьшаем S . По ходу у нас S на единицу.

Далее находим $\log_2 p$ и a и проверяем первое условие. Для этого храним 1 переменной флаг, значение которой устроит на false, если условие не выполнено. Если флаг $==$ false, выходим из программы, выводим сообщение. Для проверки второго условия создаем список $\{0, S-1\}$ и проверим все условия для всех v . Если для 1 верно, то условие прошло. Выводим, что оно верно.

Олимпиада школьников «Надежда энергетики»

№ группы

№ группы

Вариант № 7102

шифр

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ЩЕВЬЁВА

ИМЯ НАДЕЖДА

ОТЧЕСТВО СЕРГЕЕВНА

Дата рождения 18.08.1998

Класс: 10

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 5 листах

Дата выполнения работы: 27.02.15
(число, месяц, год)

Подпись участника олимпиады:

Щев

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



№3

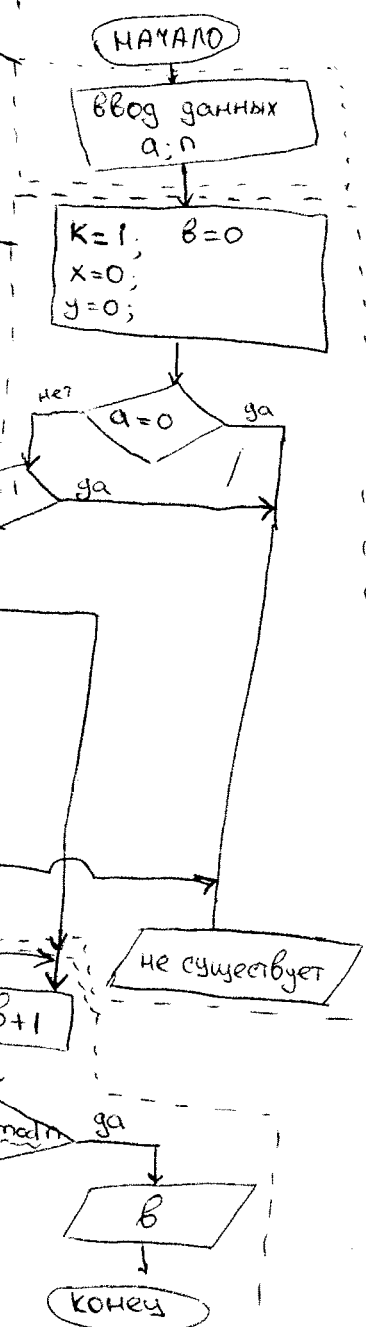
Видан доп. лист №1 Видан доп. лист №2

1) Ввести данные (в качестве примера приведу схему)

2) Проверить: являются ли числа a и n взаимно простыми. 3) Если a и n не взаимно просты, то по условию мультипликативно обратного числа не существует (\Rightarrow результат: не существует).

Если a и n взаимно просты, то мультипликативно обратное число существует по условию. Найдем его, проверяя все значения от 0 до $(n-1)$, т.к. по условию $\exists v, \text{т.к. } (a, n) = 1 \text{ и } 0 \leq v < n$

* \Rightarrow взаимнопросты a и n

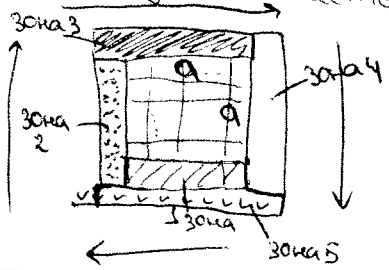


комментарий к пункту 3:
 $v \neq 0$, т.к. иначе $a \cdot v = 0$,
 и значит $a \cdot v \equiv 0 \pmod n$



№4

Заметим, что для k -ого квадрата нужно 3 рейки, если он граничит с одним уже имеющимся, и 2 рейки, если он граничит с двумя уже имеющимися квадратами. Тогда, если уже имеется квадрат $a \times a$ квадратов, то (стороны квадраты (где $a \geq 2$) по стрелкам):



- в зонах 2 и 3 „а”-квадратам нужно 2 рейки, а одну (первую) нужно 3 рейки
- в зонах 4 и 5 (а-квадратам нужно 2 рейки, а одну (первую) нужно 3 рейки для построения см продолжим на месте 02

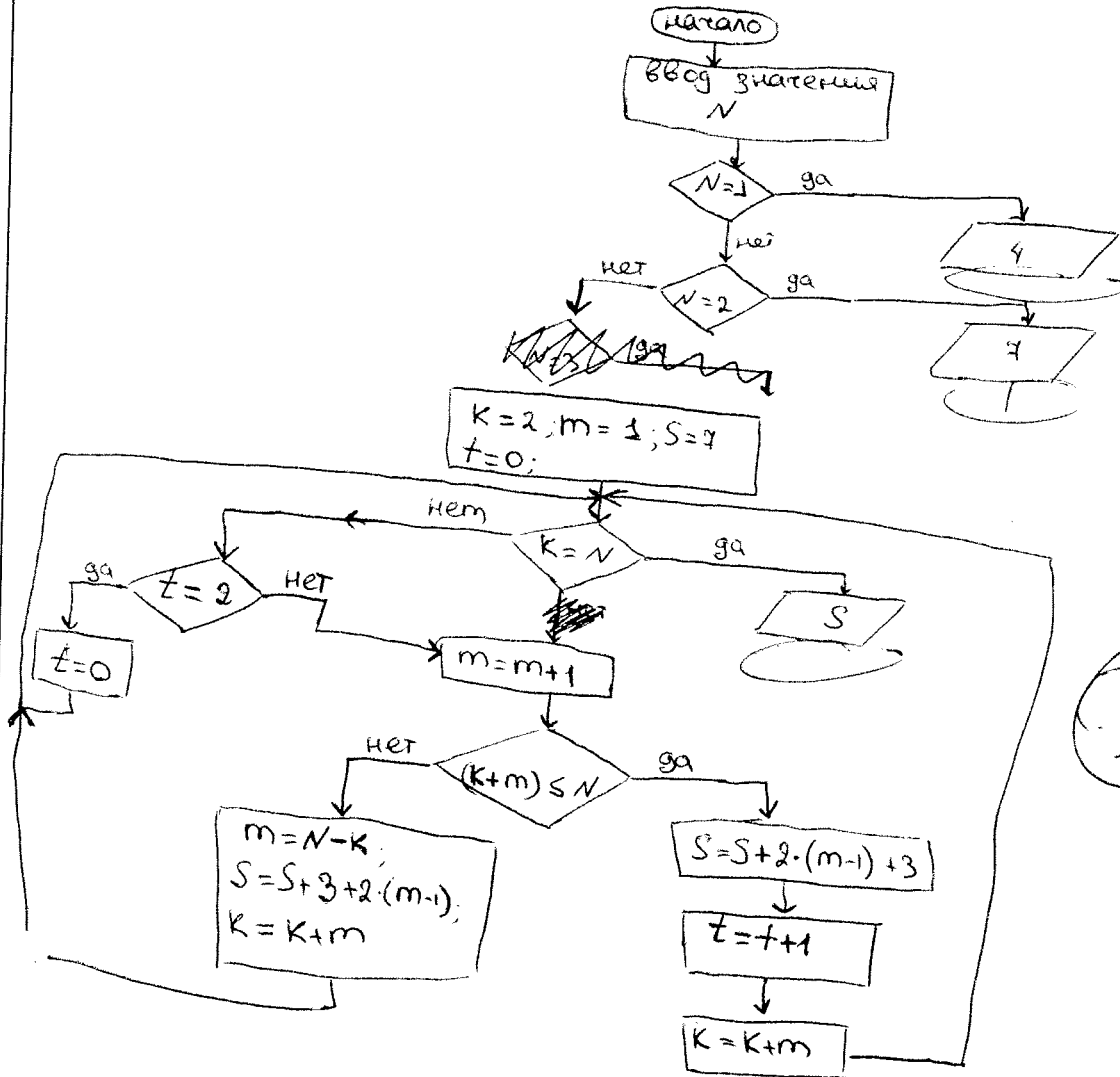


№4 (продолжение)

Тогда заметишь последовательность кол-ва нужных реек для каждого следующего квадрата:

4; 3; 3; 2; 3; 2; 3; 2; 2; 3; 2; 2; 3; 2; 2; 2; ...
 $3; 1 \times 2$ $3; 1 \times 2$ $3; 2 \times 2$ $3; 2 \times 2$ $3; 3 \times 2$ $3; 3 \times 2$ $3; 4 \times 2$ $3; 4 \times 2$
кол-во звеньев

Исходя из этого привожу пример алгоритма, считающего нужное кол-во реек



№2

Двоично-десятичный код удобно использовать, когда ~~десятичные числа~~ ~~нужно~~ нужно работать с десятичными числами, записанными в системе счисления типа римской - ~~когда эти числа не соответствуют~~ - непозиционных с.с. (кажется, они называются так)

Также такая система счисления удобна, в случае выполнения действий с многоразрядными числами (например, более 4-разрядов) без использования компьютера. (см продолжение на листе 03)



N2 (продолжение)

... III. тогда не нужно производить сложный перевод в двоичную с.с. (при этом приходится выполнять большое кол-во делений \Rightarrow большая вероятность допустить ошибку), достаточно переводить все цифры. (при этом в двоично-десятичном коде также можно выполнять алгоритмические действия)

Двоично-десятичную с.с. удобно использовать, если необходимо знать кол-во символов в десятичной записи.

N1

I перевод из восьмиричной с.с. в десятичную с.с.

1) посчитать количество знаков в записи числа A.

2) посчитать чему равно пример:

число A в десятичной с.с. (B):

$$B = (A \bmod 10^0) \cdot 8^0 + (A \bmod 10^2) \cdot 8^1 + \dots + (A \bmod 10^{n-1}) \cdot 8^{n-1}$$

II перевод из десятичной с.с. в римскую с.с.

3) $X_1 = B \bmod 10^3$

4) $str_1 = \underbrace{MM \dots M}_{X_1}$

5) $X_2 = (B - B \bmod 10^3) \bmod 10^2$

6) $X_2 < 4$, тогда $str_2 = \underbrace{C \dots C}_{X_2}$

~~$4 < X_2 < 8$, тогда $str_2 =$~~

$X_2 = 4$, тогда $str_2 = CD$

$X_2 = 5$, тогда $str_2 = D$

$5 < X_2 < 9$, тогда $str_2 = \underbrace{DCS}_{X_2-5}$

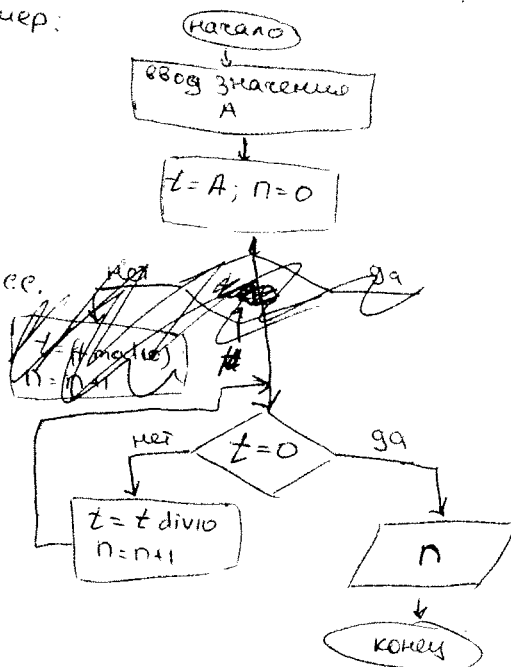
$X_2 = 9$, тогда $str_2 = CM$

7) $X_3 = (B - B \bmod 10^2) \bmod 10$

8) $X_3 < 4$, тогда $str_3 = \underbrace{X \dots X}_{X_3}$

$X_3 = 4$, тогда $str_3 = XL$

~~$5 < X_3 < 9$, тогда $str_3 = \underbrace{LX \dots X}_{X_3-5}$~~



см. продолжение на стр 04



N1 (продолжение)

Вывод дан. лист N1
И
Лешков

8) $x_3 = 9$, тогда $str_3 = XC$

9) ~~$x_4 = B \pmod{10}$~~

$x_4 = B \pmod{10}$

10) $x_4 < 4$, тогда $str_4 = \underbrace{I \dots I}_{x_4}$ • $x_4 = 4$, тогда $str_4 = IV$

• $4 < x_4 \leq 8$, тогда $str_4 = \underbrace{V I \dots I}_{x_4 - 5}$

• $x_4 = 9$, тогда $str_4 = IX$

11) $str_n = str_1 + str_2 + str_3 + str_4$

12) вывод ответа: str_n .

Результаты хранить в ^{N5} двух таблицах.

Алгоритм решения задачи

1) в табл. 1. третьей столбцу („результат“) упорядочить по убыванию значений строки n

2) для каждого значения A третьего столбца по порядку (сверху вниз) проделать действия (в Excel, например, можно написать формулу):

1) в таблице два найти значение в третьей столбце, равное значению A (не искать начиная с первой строки. В случае нахождения, остальные строки не рассматривать). Пусть найденное значение в строке k.

2) значение первого и второго столбца строки k второй таблицы, перенести в первый и второй столбцы строки n первой таблицы соответственно.

3) значения строки k второй таблицы убрать (не убирая строку). Чтобы в случае равных результатов избежать повторения строк.

N6

В решении задачи будем считать, что данные записываются в строку т.е.

N6p

N1	Р03-71	Р03-72	N2	Р03-71	Р03-72	...
----	--------	--------	----	--------	--------	-----

см. продолжение на листе 05



N 6 (продолжение)

Выдан деп. лист N 2

I проверить все ли данные поместимся,
если кол-во сыгранных партий x

1) если $3x > M \cdot N$, то не все данные есть в таблице \Rightarrow
 \Rightarrow ответить на вопрос не получится

~~II посчитать кол-во~~

Рассмотрим несколько вариантов (M -кол-во столбцов).

а) $M \equiv 0 \pmod 3$

~~I~~ II Найдём сумму очков, набранных Петром

2) просуммируем значения в строке столбцах i ,
где $i \equiv 2 \pmod 3$; результат S_1

III Найдём сумму очков, набранных Даниилом

3) просуммируем значения в столбцах j , где $j \equiv 0 \pmod 3$
Результат S_2 .

б) $M \equiv 1 \pmod 3$

II Найдём сумму очков S_1 , набранных Петром

~~для~~ для строк $1; 2; 3$ и $4; 5; 6$ и $7; 8; 9 \dots M-2; M-1; N$:

2) Найдём сумму

$$S_1 = (1; 2) + (1; 5) + \dots + (1; M-2) + (2; 1) + (2; 4) + (2; 7) + \dots + (2; M) + \\ + (3; 3) + (3; 6) + \dots + (3; M-1)$$

II Найдём сумму очков S_2 , набранных Даниилом

$$S_2 = (1; 3) + (1; 6) + \dots + (1; M-1) + (2; 2) + (2; 5) + \dots + (2; M-2) + \\ + (3; 1) + (3; 4) + \dots + (3; M)$$

в) $M \equiv 2 \pmod 3$

Аналогично пункту б) разобьём таблицу на таблицы $M \times 3$

$$II S_1 = (1; 2) + (1; 5) + \dots + (1; M) + (2; 3) + (2; 6) + \dots + (2; M-2) +$$

$$+ (3; 1) + (3; 4) + \dots + (3; M-1)$$

$$III S_2 = (1; 3) + (1; 6) + \dots + (1; M-2) + (2; 1) + (2; 4) + \dots + (2; M-1) +$$

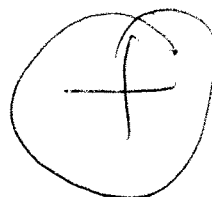
$$+ (3; 2) + (3; 5) + \dots + (3; M)$$

IV Сравним S_1 и S_2

• $S_1 > S_2 \Rightarrow$ ответ Данил

• $S_2 > S_1 \Rightarrow$ ответ Петр

• $S_1 = S_2 \Rightarrow$ ответ никто



Олимпиада школьников «Надежда энергетики»

[Empty box]

№ группы

Вариант № 7092

ZE 51-28

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ЯРАБАЕВА

ИМЯ Юлия

ОТЧЕСТВО ЕВГЕНЬЕВНА

Дата рождения 28.06.1989

Класс: 9 Б

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

Дата выполнения работы: 27.02.2015
(число, месяц, год)

Подпись участника олимпиады: 

Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



3) Первое что нам необходимо сделать, две вещи это узнать, сколько цифр в числе p . Затем найти количество остатков от деления на p , вычислив простую a p . Далее вывести это количество.

1. Считаем число p (на языке программирования Паскаль $\#$: $\text{read}(p)$)
 2. Запомним, что ~~количество остатков~~ при делении числа a на число b меньше чем b и меньше или равно a . (т.е. все возможные остатки, или ~~число b число на которое делим~~). Знаем мы можем ~~рассмотреть~~ ^{проверить} числа от 1 до p (не включая p), без других возможных остатков для не может.

Если число от 1 до p является ^{взаимно} простым к p , то это и есть один из ^{возможных} остатков при делении на p , вычислив простых a .
 Знаем мы досконально ^{узнав список} ~~проверить~~ ^{вперед} ~~или~~ ^{однажды} простое число к p .
 число от 1 до p

Для этого применим алгоритм Евклида.

На языке программирования Паскаль это будет выглядеть так:

```

k:=0; y:=p; j:=1
for i:=1 to p-1 do
begin
while (j<>y) do
if (y>j) then y:=y-j else j:=j-y;
if (j=1) then k:=k+1;
j:=i+1; y:=p;
end;
write(k);
end.
  
```



где k подсчитывается количество остатков.

5) Нам нужно найти всевозможные варианты индекса, т.е. числа n которое больше 9999 и меньше 100.000.000 (по условию полевой индекс содержит от 5 до 8 знаков) и делится на 7 и 11. Но 7 и 11 это взаимно простые числа, значит число которое делится на 7 и 11, делится и на 77 ($11 \cdot 7 = 77$). Первым ^(минимальным) таким числом будет 10010. Следующее число будет больше предыдущего на 77.





1. Находим наименьшее число, удовлетворяющее условию.
 Для $z=0$ 10.000 делим на 77 . Получим примерно 130 . Далее ~~с наименьшим~~ ^(наименьшее) ~~с наименьшим~~ ^{число} умножим 77 на 130 и получим 10010 . (1)

2. Будем выводить на экран последующее число прибавляя к предыдущему 77 , пока ~~эта~~ ~~это~~ число не станет больше или равно ~~(10)~~ $9.999.999$.

На языке программирования Паскаль это будет выглядеть так:
 $b := 9999$
 while $(k < 99.999.999)$ do
 begin
 $b := b + 77$
 write(b)
 end;

Мы пишем ~~мы пишем~~ $(k < 99.999.999)$ в.к.

Если мы прибавим к $99.999.999$ 77 то получим $99.999.999$, и условие будет соблюдено. (1)

(2) Я считаю что самый удобный способ хранения данных о часовых поясах стран, это хранить данные в таблицах. Создадим базу данных, где будут содержаться 2 таблицы. В 1-ой будут содержаться данные по странам, которые не переводят время, а во 2-ой по тем, кто переводит время. (Так мы избежим повторений). ~~(Содержим для таблицы будут содержать следующие данные: часовые)~~ В эту таблицу будут занесены списки всех стран. (в.к. нам будет задан часовое пояс, указывать его в таблице не нужно).

~~Когда появляется запрос, мы~~

При определении местного времени, первое что мы делаем, это определяем запрос, на основе сплюснотью которого находим заданную страну. Если она находится в таблице $\neq 1$, то мы считаем местное время, прибавив к часам по Гринвичу то число, что стоит после GMT в заданном часовом поясе.



Если же сбрана переводит время, то местное время мы считаем прибавив к часам по Гринвичу числовое значение после GMT в заданном часовом поясе и в зависимости от ~~времени~~ даты сегодняшней прибавим или убавим 1 час.

④ 1. Символьная строка m и n ($read(m;n)$)

2. Делаем M раз: $k = k \cdot M$; где k это число которое подставляется M .

3. Выводим k . ($write(k)$).

Т.к. строка получается достаточно большой при составлении программы в Паскале, необходимо ~~здесь~~ ~~longint~~ записать M и k , как $longint$. (~~то longint не влезет~~)

⑦ 1. Посчитаем факториал $44!$.

Т.е. на языке Паскаль это будет выглядеть так:

$k := 1$
for $i = 1$ до 44 do
 $k := k \cdot i$;

где k подставляет значение $44!$

2. Запишем это число как строку. $a(string)$.

3. Проверим каждый элемент этой строки, и если он равен "4", то выведем ~~это~~ число, равное

длине этой строки минус порядковый номер ~~этого~~ символа, равного "4", и прибавим 1.

Т.е. $if\ a[i] = '4'$ then $l := length(a) - i + 1$
 $write(l)$
end.

① Я считаю, что эта система числения является позиционной, т.к. ~~каждый~~ разряд числа не имеет свое значение.