

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 711205 61-73

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ

АБРАМОВ

ИМЯ

Дмитрий

ОТЧЕСТВО

ПАВЛОВИЧДата  
рождения21.01.1998Класс: 11

Предмет

ИнформатикаЭтап: ЗаключительнойРабота выполнена на 5 листахДата 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

Bogart, Inc., Inc.

1) представить данное число в виде суммы первых же трех его разрядов. Например:  $521 = \underline{500} + \underline{20} + 1$  сотен десятков единиц.

$$9\ 383 = \underbrace{9000}_{\text{тысяча}} + \underbrace{300}_{\text{сто}} + \underbrace{80}_{\text{восьмидесять}} + \underbrace{3}_{\text{три}} \quad \begin{matrix} \text{тысяча} \\ \text{сто} \\ \text{восьмидесять} \\ \text{три} \end{matrix}$$

$$\text{Например: } 521 = 500 + 20 + 1 \\ \text{или: } 500 \rightarrow \textcircled{D} \rightarrow \textcircled{D} \\ 20 \rightarrow \textcircled{X} \rightarrow \textcircled{DXX} \\ 1 \rightarrow \textcircled{I} \rightarrow \textcircled{DXXI}$$

1

3) Теперь обсудим сам процесс перевода:

$$\text{Mr. green, ems} \quad I = 1 \quad L = 50 \\ V = 5 \quad C = 100 \quad M = 1000 \\ X = 10 \quad D = 500$$

Please access (paywall) my site below N.

глумно скажем такую букву Р. с., потому  
система равна числу  $N$  (единичные единицы)  
тогда ~~формированием~~<sup>суммированием</sup> эту букву, а из  $N$  корреляций  
закартируем таки буквы в 10-ой СС наблюдения  
так, пока  $N > 0$ .

мо, нока N>> : Могутли: Есле  $N =$  (буква більшого розміру -  
- буква меншого розміру), то в замісочній місці -  
- то маємо (менша буква перед більшою).

$$\text{Hampshire} : 400 = 100 + (100 + 100 + 100) - \text{all} .$$

$$400 = (500 - 100) = \text{CD} .$$



м.е. если в записи числа (римской с.с.)  
буква, у которой значение меньшее стоят  
после буквой, у которой значение большее ( $\chi$ ),  
 то надо из удал. большей записать меньшее,  
 а если наоборот ( $\chi$ ), то оставить.

№2

Когда запись расположено число в  $(2-10)$  с.с.  
 чиселодробнее тогда, когда или не запись  
 общей числоряднее, которая эта запись  
 записывает. Весь в десятичной системе счисл.,  
 запись (числа) числа в 2-ой с.с. меньше,  
 чем запись (числа) в  $(2-10)$ -ой.

$$\begin{aligned} i_1 &= \log_2 a & i_1 < i_2, \text{ где } a - \text{коэффициент } b \\ i_2 &= \log_2 (b \cdot u) & \uparrow \quad \uparrow \quad \text{запись числа в 2-cc.} \\ & \quad \text{двоич.} & \quad \text{запись } b - \text{коэффициент } b \text{ записан} \\ & \quad \text{запись} & \quad \text{в } 10-\text{ой с.с.} \end{aligned}$$
7

Например;  $311_{10} = 100110111_2 = 0011_2, 0001_2 0001_{2-10}$ .

$$\begin{array}{l} a=9 \\ b=4 \cdot 3=12 \end{array} \quad \log_2 9 < \log_2 12.$$

Однако; • в случае, когда нет записи коэффициентов.  
 • в 2-10 с.с. нет записи операций  
однозначных ( $+, -, \cdot, :)$

№3.

1) Вводим ( $n$ ) и ( $k$ ).

2) проверяем, являются ли они взаимно простыми (если нет, то вводим "Не существует", конец программы).

Если являются, идем дальше.



3) проверяем в от  $\Phi_{go}(n-l)$ .  $(a \cdot b \bmod n = 1)$   
 проверяется выражение  $a \cdot b \equiv 1 \pmod{n}$ .  $\frac{a \cdot b}{n} = \dots$ , ①  
 если такое в существует, то бывает ли это,  
 если нет  $b = n-1$ , а результат не  $1$ , то  
 бывает ли это существует".  
 Конец.

(+)

нч

1) Вводим  $N$ -количество квадратов.

2) если  $N = l$ , бывает ли  
 если  $N = 2$ , бывает ли

3) если  $N > 3$ .

$N := N - 2$  (м.к. 2 квадрата вдали в ~~тесноте~~  
 нужно  $\sqrt{2}$ )

$k := l$  (как-то есть нужно представить в виде  $\sqrt{2}$ )

$a := l$  (сколько раз нужно пройти вперед перед  
 поворотом). (сколько квадратов перед поворотом).

Умакс на первом же я бывало можно -

поскольку:  $(3 + 2a + 3 + 2a) -$  как-то есть, после

которого  $a$  увеличивается на  $l$ . ( $a+1$ ).

м.к. как-то квадратов  $= (l + 2 + 1 + a) = (2a + 2)$

4) Если  $N - (2a + 2) > 0$ , то

$$\underline{N := 2a} \quad N := N - (2a + 2);$$

$$k := 3 + 3 + 2a + 2a;$$

$$a := a + 1.$$

повторное условие (4) пока это возможно

5) если  $N > 0$  ~~то~~ ~~значит~~ ~~противо~~.

если  $N > 0$  ~~то~~ ~~значит~~ ~~находи!~~

$$N := N - 1;$$

$$k := k + 3;$$

$$a := a + 1$$

$$N := N - 1$$

(предоставлено на  
 другом листе)



Бычко Дон. Иван  
7  
Башк

5) если  $N=0$ , то конец программы.

6) если  $N > 0$ , то

$$N := N - 1;$$

$$K := K + 3;$$

• повторить а раз пока  $N > 0$

$$N := N - 1;$$

$$K := K + 2;$$

⊕

• ~~зану  $N > 0$~~

$$N :=$$

7) выполнить цикл (6) пока  $N < 0$ .

8) вывести  $K$ .

№ 5

1) как хранить?

записи в массиве (2 числовых и 1 символов)

в массиве a - храним  $n \pi / k$

b - значение параллеп

c - результатом.

2) как работать?

загад: упорядочение по возрастанию  
значений результата.

Значит, будем упорядочивать массив

c, а основное массива ( $a, b$ ) просим  
дублировать.

3) как упорядочить массив с по возрастанию  
по указано ви выше № 5.



3) Упорядочить массив с (результатом) ~~буквой~~ <sup>буквой</sup> №  
по возрастанию.

Женя

берём первый элемент и сравниваем  
с остальными. Если какой-то меньше,  
меняем его, то меняем их местами.  
Берём 2-ой элемент и сравниваем его  
со следующим, поменяй 3-ий и  
так далее до конца массива. ↑  
Не забываем в это время  
изменять массивы а и в.  
В результате введем все массивы.

№ 6

Составим сумму всех парных  $y$   
каждого игрока.

$y_{\text{тёнка}} = \underline{\text{rest}^1}$  (за все парные) <sup>один</sup>

$y_{\text{Данила}} = \underline{\text{rest}^2}$

если  $\text{rest}^1 > \text{rest}^2$  введем  $\text{tёнка}$   
если  $\text{rest}^2 > \text{rest}^1$  введем  $\text{Данила}$   
если  $\text{rest}^1 = \text{rest}^2$  введем  $\text{ничья}$ .

7

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7092

*ZE 51-26*

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ Андреева  
ИМЯ Влада  
ОТЧЕСТВО Александровна  
Дата рождения 04.08.1999 Класс: 9  
Предмет информатика Этап: заключительный  
Работа выполнена на 3 листах Дата выполнения работы: 27.02.2015  
(число, месяц, год)

Подпись участника олимпиады:

*Влада*

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.

Код 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. Код 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 m do 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]; d := 0;
  for i := 1 to m do
  begin
    d1 := d;
    d := 0;
    while (d1 <> 0) do
```



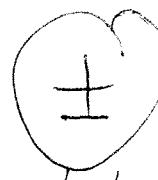
```
begin
  c := c + (a[j] * b[i]);
  a[j] := c mod 10;
  c := c div 10;
  d1 := c;
  d := d + 1;
end;
end;
for i := d downto 1 do write(a[i]);
end.
```

№5. Код C++

```
#include <iostream>
```

```
using namespace std;
```

```
int main(){
  for(long int i = 10000; i < 100000000; ++i){
    if ((i & 7 == 0) && (i & 11 == 0)) cout << i << '\n';
  }
  return 0;
}
```

№7. Код Pascal.

```
var a: array[1..10000] of longint;
  d, c, i, j: longint;
```

```
begin
```

```
  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.

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

Факториалы мы назовем множителями.

⊕

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

№2.

↑	↑	↑
1	2	3

1. Переводят ли в данной системе времена?
2. Сколько один из дней с зимним временем?
3. Сколько один из дней с летним временем?

⊖

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

№6.

При расчете на компьютере складываются двоичные представления чисел.



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112  
JM 25-46

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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: 24.02.2016  
(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. ( $I, V, X, L, M \dots$ ) (⊕)

анализируя, для чисел  $< 4$  число заменяется с помощью римской буквы "I", где число  $< 4$  число заменяется с помощью римской буквы "I", где число  $= 4$   $\Rightarrow$  число сначала "I", затем "V" т.е. число  $i = 1$  и  $v = 5$ .  
 сколько же будет число  $go$ ?  
 также для чисел заменяется  $V$ , а дальше столбце "I", сколько с.с. есть  
 написано? где число  $5$ ?  
 дальше где число заменяется  $X$ ? в таком кол-ве  
 а затем  $X$ ? где  $X$ ? где  $> 5$  будет в отрицательном кол-ве  $10$ ?  
 нужно ли это число и  $10$  (сколько заменяется "I")?  
 где разрядов будет включено след-ее: если равно цифра в  
 первом, сколько в замененном числе будет цифрой  $X$ ? и так далее.  
 $i = 1, v = 5, x = 10, l = 50, M = 100 \dots$

2) число  $311_3$ : в д.с.с.:  $311_{10} = 1.0011.0111_2$

(т.к.  $311_3$

$$\begin{array}{r} 155 \text{ L} \\ \hline 0 \quad 44 \quad L^2 \\ \quad 0 \quad 38 \quad L^2 \\ \quad \quad 0 \quad 70 \quad L^2 \\ \quad \quad \quad 0 \quad 9 \quad L^2 \\ \quad \quad \quad \quad 0 \quad 4 \quad L^2 \\ \quad \quad \quad \quad \quad 0 \quad 2 \quad L^2 \\ \quad \quad \quad \quad \quad \quad 0 \quad 1 \end{array}$$

в д.с.с.

(⊕)

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

- анализируя двоично-десятичной схемы числа членообразно, если:
- 1) На каждую цифру числа в двоичной (в четырех) форме
  - 2) Если каждая цифра исходного числа, к которой нужно привести эту цифру, чтобы первое из д.с.с в 4-й с.с
  - 3) Переведенное число необходимо перевести в д.с.с в 4-й с.с
  - 4) Число членообразное решать по разрядам.

3). Введенное число число  $a$  ( $0 \leq a < n$ ) и значение модуля  $B$ . Проверка, является ли число  $a$  и  $B$  числом членообразным. Для этого проверка, является ли число  $a$  и  $B$  числом членообразным  $(2, 3, 5, 7, \dots)$ . Если это условие выполняется и исходное число  $a$  и  $B$  членообразные, то можно выполнить деление:  $(n - B) \bmod a = 0 \Rightarrow n - B = a \Leftrightarrow B = n - a$

иначе членообразное обратное число  $a$  не существует. т.к. для данного числа  $a$  не существует  $B$  (⊕)





④. Ихим N-кок-бо квадратов. ( $N: \text{integer}, N > 0$ )

Создать массив  $a[1..N]$ :

S-кок-бо рекк;  
Изначально,  $S=4$ . if ( $i=1$  and  $(N=1)$ ) then  $S=4$ ,  $W=N$  ( $S$ );  
for  $i:=2$  to  $N$  do begin  
if (( $i=2$ ) or ( $i \bmod 2=1$ )) then  $S:=S+3$ ;  $\frac{+}{\text{if}}$   
if ( $i \bmod 2=0$ ) and ( $i > 2$ ) then  $S:=S+2$ ;  
end;

Прим: программа Pascal S-менинг. Ихим рекк, Ихимдик-код  
Шабака:  $N=5$ ;  $i=1: S=4$ ;  $i=2: S=7$ ;  $i=3: 3 \bmod 2=1: S=10$ ;  $i=4: 4 \bmod 2=0: S=12$   
 $i=5: 5 \bmod 2=1 \Rightarrow S=15$  (киш үзбек-ер даан-тас төрлини)

⑤. Альбомнико сордат базу данных між в программе M-Excel,  
між в программе MS Access.

Заполнить форму так, как сказано по услою  $\ominus$ .  
Заполн, введиши символ "результат", ии зада в настрой  
базы данных, сордат упорядочить по возрастанию  
(от 1-го к последнему). Ури таш, закрепив весь строку  
форме, читай. Использовалась не одна строка в базе,  
а все строки ( $A_1$ , начиная результат).

Лен сордат двухмерный массив в программе Pascal  $a[i,j]$   
i- строка матрицы  $a[i,1] - A_{1,1}$ ;  $[i,2] - A_{1,2}$  - заполнение  
ii-я строка функции assign, reset и применение op-ю stl (использование  
 $[i,3]$ -шабака результатов (заполнение аналогично)  
Задал шабака программы ии матрицы for  $i:=2$  to  $N$   
( $N$ -номер ии строки): if  $a[i,3] < a[i,2]$  then begin  
 $a[i,3]:=a[i,2]$ ;  $a[i,1]:=a[i,2]$ ; end;  
такши от排序, маркеру производим результат. Использовалось  
производим сортировка с  $i=1$  до  $i=N$ .

⑥. Т.к. по условиям задачи (заполненная) табл не дана,  
согласно предложенному ее салле:

Давим синтакс, чио на синтаксе 3 символа  
( $No$ ,  $res-1$ ,  $res-2$ ) а кок-бо строк =  $M$ .  
Давим избегнини ии, недостаточно знать, кел результат  
 $\sum_{k=1}^M$  (данни-2). Ирга, скончательно, результат яхра  $\frac{+}{\text{if}}$   
 $P=\sum_{k=1}^M X(M,k)$ , i.e. скончательно весь результат скончательно  $M$ .

а результат данни:  $D=\sum_{k=1}^M Y(M,k)$ ; (сумма весь келек о.03)  
В касе избегнини тот, кио  $\sum_{k=1}^M$  келек дөзвле орек. Ели  $P>D$   
согласно write ('Данни'), а ели  $D>P \Rightarrow$  write ('Данни'); ии согласно  $P=D$  writer 'Каса'.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 711205 61-82

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ

Баранов

ИМЯ

Юрий

ОТЧЕСТВО

Александрович

Дата

рождения

22.12.97Класс: 11

Предмет

информатикаЭтап: заключительный

Работа выполнена на

7 листахДата 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.



№6.

безал 1 доп.мест + 1 доп.мест + 1 доп.мест + 1 доп.мест *зар*

Создание 2 числа для хранения суммарных очков  
Петя и Данила соответственно:

(Var

P, q : longint; 0..1000, 1..3

a: array [ 1..3, 0..1000 ] of integer; - блокнот строков  
integer;

нагадо

Нужен цикл вниз по блокноту и присваиваете очки:

n - высота страницы, m - ширине стр.

for i := 1 to n do

begin

p := p + a[ i, 2 ]; // прибавляем a[i, 2] и a[i, 3], потому  
q := q + a[ i, 3 ]; // что в a[i, 1] записан номер персон.

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

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

end;

if p &gt; q then

writeln ('Петя');

if p &lt; q then

writeln ('Данила');

if p = q then

writeln ('ничья');

конец.

№7. i, n: integer;

begin

writeln (n);

for i := 1 to (n div 10) do

write ('x');



IV5.

Быстрее всего было бы создать 3 массива где хранятся экспериментов (num, param, res) и один временный массив, в который записать числа от 1 до n (номера экспериментов), а дальше отсортировать массив  $a[i]$  по возрастанию, сравнивая  ~~$a[i]$~~   $res[a[i]]$  и  $res[a[j]]$ , любой сортировкой за  $O(n \log n)$ , к примеру, шелл-сортировкой.

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

One удобнее сделать процедуру обмена swap.

Type

```
point = ^ list - указатель на начало списка
list = список;
num: integer;
param: string;
res: integer; n: point;
end; var
```

†

```
procedure swap(p, q: point);
```

```
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;
```

конец.

```
var b, q: point;
```

```
flag: boolean;
```

начало

```
b := nil; l := b; // b - начало списка
```

если остаток не пуст, то

начало

создать новый элемент(q);

ввод(q^.num, q^.param, q^.res); ввод из остатка

```
l := q;
```

```
l^.next := q;
```

конец.

↓ продолжение на 4 листе



Xt.

```

s : string;           // pesyiatat
n : integer;          // runcu
i : integer;
var
    s : array[1..4] of char;
    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';
    if n mod 10 = 9 then
        s := s + 'IX';
    if (n mod 10 >= 5) and (n mod 10 < 3) then
        var
            s := s + 'V';
            for i := 1 to (n mod 10 - 5) do
                s := s + 'I';
    writeln(s);
    begin
        writeln(s);
    end.

```

ВВОД ДАННЫХ ИЗ ФАЙЛА 

Пока flag не будет делать  
нр

создать (q);

ввести (q<sup>1</sup>.num, q<sup>1</sup>.param, q<sup>1</sup>.res);

l<sup>1</sup>.n := q;

l := q;

нр;

flag := true;

~~while flag~~

пока flag делает

нр

flag := false;

l := B;

пока l<sup>1</sup>.n <> nil делать

нр

if l<sup>1</sup>.res > l<sup>1</sup>.n<sup>1</sup>.res then

нр

swap (l, l<sup>1</sup>.n);

flag := true;

нр

l := l<sup>1</sup>.n;

сортировка

|| если был совершен обмен,  
стартовали флаги, чтобы  
сортировка продолжалась

~~нр;~~ l := B; ~~while~~ ПОКА (<>nil делать нр ввести (l<sup>1</sup>.param), l := l<sup>1</sup>.n; ~~нр;~~

нр.

N2.

Чтобы ответить на вопрос № 15 необходимо, поскольку если записывать каждую цифру 10-тибитного числового отображения 4 битами, это приведет к существенным затратам памяти.

Можно предположить, что удобно использовать данный подход если требуется максимальное быстрое узнавание любого цифру числа. Тогда для этого не нужно будет каждый раз переводить число, после вычислений, в привычную.



№3.

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

Var

a, b, k, n: integer;

flag: boolean; // flag-метка не найдено ли значение

k:=0

while ( $\frac{nk+1}{a} < b$ ) do if flag

begin

k:=0; ввести(n, a); flag:=true;

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

end

если  $(nk+1) \bmod a = 0$  то

begin

flag:=false;

b:=(nk+1) div a;

k:=;

k:=k+1;

end;

если flag ТО

вывести('не существует')

иначе

вывести(b);

end.





N1.

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

• var n, t, i : integer

WR блеск(n);  
for i := 1 to n div 1000 do  
    write('M');  
    n := n mod 1000;  
    t := n div 100;  
    if t = 3 then  
        write('CM');  
    if (t ≥ 5) и (t < 3) then  
        WR  
        write('D');  
        for i := 1 to (t mod 5) do  
            write('C');  
        KIS;  
        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 = 3 then write('XC');  
            if (t < 3) и (t ≥ 5) then  
                WR  
                write('L');  
                for i := 1 to (t mod 5) do write('X');  
                KIS;  
                if t = 4 then write('XL');  
                if t < 4 then  
                    for i := 1 to t do write('X');  
                    t := n mod 10;  
                    if t = 3 then write('IX');  
                    if (t ≥ 5) и (t < 3) then  
                        WR  
                        write('V');  
                        for i := 1 to (t mod 5) do write('I');  
                        KIS;  
                        if t = 4 then write('IV');  
                        if t < 4 then  
                            for i := 1 to t do write('I');

PASPA COTEN

SECRETKOB

equally

KIS





N4.

by

Учитывая небольшое ограничение на  $N$ , можно вспомнить о массиве.

var

a: array [1..16000] of integer;

i, j, k, n: integer;

begin  
writeln b6og(n);

a[1]:= 4;

a[2]:= 7;

a[3]:= 10;

a[4]:= 12;

i:= 4; k:= 1;

~~while i < n do~~

ПОКА (i &lt; n) делатъ

и

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;

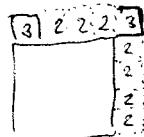
k;

writeln(a[n]);

кн.

~~Решение:~~

Решение:



Идея алгоритма в том, что панелька половины "оборота" вокруг квадрата робот тратит столько 3 ребра, затем  $k$  раз тратит 2, затем опять 3 и погоду  $k+1$  раз 2 ребра. Соответственно  $k$  с каждой разой увеличивается.

## Олимпиада школьников «Надежда энергетики»

BC 28-34

← Не заполнять  
Заполняется  
ответственным  
работником

№ группы

Вариант № 7/13

шифр

ФАМИЛИЯ БАСАРГИНА

ИМЯ СВЕТЛНА

ОТЧЕСТВО НОРЬЕВНА

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:

бас

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



## Олимпиада школьников «Надежда энергетики»

Вариант: 7/13

ШИФР, НЕ ЗАПОЛНЯТЬ! ⇒

BC 28-34

№1 Для перевода списка надо необходимо восстановить последовательность Роджерса до заданного  $N$  (для этого ищем используемый список, т.к. не знаем сколько будет переменных; в данном списке можно перейти как к предыдущему элементу так и к следующему)



## Алгоритм Перевод;

## Переменные

$N$ : целое; //  $N$ -заданное число; а в необходимо  
 $a, b$ : натуральные; где создание поиска т.к. постепенно между  
 возрастает определяющее значение,

Нагаю

$b_{bog}(N)$ .

each ( $N < 0$ ) mo

~~Бород~~ бород ("значение руки неизвестно");  
~~Бород~~ бород (Бород);

case  $\nabla(N=0)$  and  $(N>1)$  no border  $(N, \geq, N)$ ,  
 case  $(N>1)$  no

создать список;  
 первої змінити списка = 1;  $a = 1$ ;  
 создать список;  
 первої змінити списка = 2;  $b = 2$ ;  
 пока  $(b < N)$

Мы  
создать новый элемент;  
новый элемент =  $a + b$ ;  
 $a = b$ ;  
 $b$  = новый элемент;

$\hat{N}$  (запись списка  $\leq N$ ) то  
 $= N = N$ -запись списка;

borbog ("10");

установлено землем.

человека к прогрессу;

переименование;

кто прими к председателю  
шаре, кто

borboog ("O"):

*ХХ ХХ ХХ  
попытка* удаётся элемен<sup>т</sup>лу снаря<sup>д</sup>; кратчай<sup>ш</sup>е предположение

Лист 0 / 1 из 0 / 5



16) Т.к. мы не знаем точное количество элементов, находясь участка, значит нам необходимо создавать список элементов а не массив

Количество списка в том, что для него не обязательно знать кол-во элементов изначально, но "иниусы", что в списке невозможно обращаться к определенному элементу, а необходимо двигаться от начала к концу.

Т.к. под условию задачи нам необходимо уметь кол-во элементов данного вида  $\Rightarrow$  расположение данного элемента нам запоминать нет необходимости а можно лишь идти по каждому саженцу и запоминать его вид.

Каждый элемент списка в нашем случае будет состоять из ~~из~~ строк в которую входит 2 элемента: 1. - кол-во ярусов (от  $M \leq N$ ) и кол-во саженцев

Пример:

Список саженцев	
кол-во ярусов	кол-во элементов
$M$	...
...	...
$N$	...
...	...

Обозначение:

список - саженец  $\Leftarrow$  (от  $M \leq N$ )

список - количество

Алгоритм Постеб:

Переменные  $M, N, K, L, P$ : натуральное;  $f$ : логическое

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  $f = \text{true}$ ;

создать список

ввод ( $C$ );

первый элемент := список\_саженец  $\neq C$ ;

первый элемент := список\_количество = 1;

пока ( $существует C$ )  $f = \text{true}$ ;

если 1 элемента списка до конца;

иначе список\_саженец =  $C$  то

если  $f = \text{true}$ ;

список\_количество = список\_количество + 1;

иначе  $f = \text{false}$  то иначе

создать новый элемент

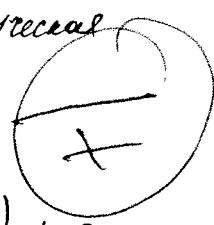
список\_саженец =  $C$ ;

список\_количество = 1

иначе ввод ( $C$ )

пока существует список иначе

вывод (список\_саженец, список\_количество);





(N6) ~~удалить элемент;~~  
к<sub>у</sub>  
конец.

(N5) Для решения данной задачи для начала можно все ~~э~~ числа заносить одномерный массив состоящий из  $N \times N$  клеток, отформировать его, а потом при выводе массива записать его в виде двумерного ( $M \times N$  клеток), при этом ~~записать~~ т.к. Алгоритм Сортировка.

Переменные  $O, N, M, a[1... 10^4 \cdot 10^3]$  целые числа  
Инициало  $j; i$ : натуральное;  
ввод ( $M, N$ );

$M$   
 $N$

если ( $M < 0$ ) или ( $N < 0$ ) то  
вывод ("Некорректные данные");

для  $i = 1$  до  $M \cdot N$

из <sub>и</sub>  
ввод ( $a[i]$ );

для  $i = 1$  до  $M \cdot N$

из <sub>и</sub>  
для  $j = 1$  до  $M \cdot N - 1$

из <sub>и</sub>  
если  $a[j] > a[j+1]$  то

из <sub>и</sub>  
~~обмен~~  $O := a[j];$

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

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

из <sub>и</sub>

к<sub>у</sub>

$k = 0;$

для <sub>и</sub>  $i := 1$  до  $M \cdot N$  делать

ввод ( $a[i]$ );

$k \leq k + 1$

если  $k = M$  то

из <sub>и</sub>  
вывод ("Enter");

$k = 0;$

из <sub>и</sub>

конец

⊕

(N4) Альгоритм Степени;

Переменные ~~и~~  
~~число~~

$A = N^M$

$i$  - натуральное;

~~вывод~~  $N, M, A$ ;

~~ввод~~

~~ввод~~ ( $N, M$ );



(14) Т.к. число  $N$  возможно представить в "двоичном" стилем  $M$ , то лучше в этом случае воспользоваться алгоритмом быстрого возведения числа в степень.

Для этого же начали ~~всегда~~ переведем степень  $M$  в двоичной вид.

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

По схеме Горнера: ~~(100111000)~~

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

Алгоритм степени;

Переменные  $k, i$ : натуральное;

$A; N, M$ : целые; а массив  $[1 \dots 1000]$ ;

начало

bbog ( $N; M$ ); или

если ( $N \leq 0$ ) || ( $M \leq 0$ ) ~~то~~ вывести ("Неверное значение")

иначе

из  $i=0$

пока  $M > 2$

из

$i=i+1$

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

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

из

$i=i+1;$

\* ~~если~~  $a[i] = 1$ ;  $A := 1;$

\* ~~если~~  $k := i \rightarrow 1$  \*

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

из

бог ( $A$ );

из

из

конец.

(\*) если ( $i > 1000$ ) или ( $N \text{ div } 10^4 \text{ mod } 1 > 1$ ) то

вывести ("Неверное значение");

из

—



(N2)

Алгоритм Поиск;  
 Переменные  $N_1; N$ : целое  
 $k$ : целое  
 массив  $a[1..50]$ ;  $\min$ : натуральное  
 начало

вывод ( $N_1$ ).

если  $N_1 < 0$  то выход ("закончите не  
 корректно");  
 где  $k := 1 \text{ до } N_1 + 50$

иначе  
 создать список;  
 новый элемент =  $k * k$ ;

куда  $\min = 100$ ;

где  $N := N_1 + 1 \text{ до } N_1 + 50$  иначе

~~подбираем сумму квадратов из списка;~~

~~ищем сумму существует ли~~  
 иначе

~~подбираем сумму квадратов из списка;~~

~~ищем сумму существует ли~~

куда  $a[N-50] = N(N-50) + 1$ ;

если  $a[N-50] < \min$  то  $\min = a[N-50]$  иначе

где  $N := 1 \text{ до } 50$  иначе

если  $a[N] = \min$  то

вывод ( $N+50; '-'$ );

иначе

конец.

(N3)

При выполнении алгоритма если заданное число  $p$  четное ( $p \bmod 2 = 0$ ), то такое число сразу можно сказать, что оно составное.  $\Rightarrow$  по данному алгоритму только нечетные числа.

~~доказательство алгоритма~~

Число  $a$  делится на 2 через член

$a$ -шестое число ( $\text{Random}(p-1)$ )

наше определение а делника осуществляется проверка числа  $p-1 = 2^s \cdot d$  по данному методу

где определяется заданное нечетное число  $a$  проверка состоит в том что если  $a^{d-1} \equiv 1 \pmod{p}$  и если эта проверка ложна то через члены от 0 до  $s-1$  проверяется условие 2. Если и это условие ложно выход "составное"

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

(F)

## Олимпиада школьников «Надежда энергетики»

<input type="text"/>	<input type="text" value="0Г 61-78"/>	← Не заполнять Заполняется ответственным работником
№ группы	Вариант №	<u>3112</u> шифр
ФАМИЛИЯ	<u>Большаков</u>	
ИМЯ	<u>Дмитрий</u>	
ОТЧЕСТВО	<u>Валерьевич</u>	
Дата рождения	<u>07.08.1994</u>	Класс: <u>11</u>
Предмет	<u>Информатика</u>	
Работа выполнена на	<u>4</u> листах	Дата выполнения работы: <u>29.02.915</u> (число, месяц, год)

Подпись участника олимпиады:

Роман

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

Задача 1базы 1 год. мес *claf*

Начало

Ввод ( $n$ ):

$$R[1] := 1000; \quad RN[1] := 'M';$$

$$R[2] := 500; \quad RN[2] := 'D';$$

$$R[3] := 100; \quad RN[3] := 'C';$$

$$R[4] := 50; \quad RN[4] := 'h';$$

$$R[5] := 10; \quad RN[5] := 'X';$$

$$R[6] := 5; \quad RN[6] := 'V';$$

$$R[7] := 1 \quad RN[7] := 'I';$$

$$i := 1;$$

Пока  $n > 0$  делай

Начало

Пока  $n \text{ div } R[i] = 0$  делай  $i := i + 1;$ 

$$p := n \text{ div } R[i];$$

если  $p \leq 3$  тогдадля  $j$  от 1 до  $p$  делай  $s := s + RN[i];$   
иначе началодля  $j$  от  $p$  до 1 делай  $s := s + RN[i+1];$ 

$$s := s + RN[i];$$

Конец;

$$n := n \bmod R[i];$$

Конец;

Вывод ( $s$ );

Конец.

 $n$  - исходное натуральное число; $P, D, I, S, RN[1..7]$  - члены шифра; $RN[1..7]$  - символы перемен; $S$  - строка;

 $\widehat{+}$

Задача 2

Очевидно, что в 2-10 системе на чисто отводится не меньше (чаще всего - больше) информации, чем в 2 системе. Но, несмотря на это, в 2-10 системе есть своя мера: ~~единицы~~, основанные на действии с отдельными цифрами десятичной записи, например - их перестановки. Такие 2-10 системы более удобны и понятны человеку, привыкшему считать в 10 системе. (✗)

Задача 3

Начало

Ввод( $n; a$ ); $c :=$  истина; $b := 2$ . $c$  - конечная переменная; $n$  - модуль; $a$  - чистое число; $b, i$  - целые числа;Пока ( $c$ ) И ( $B < n-1$ ) делать

начало

Пока  $(a \cdot b) \bmod n < 1$  И ( $B < n-1$ ) делать  $B := B+1$ ;Если  $(a \cdot b) \bmod n = 1$  ТО начало  $c := 10 * b$ ;} для  $i$  от 2 до  $\sqrt{a}$  делатьначало если  $(a \bmod i = 0)$  И  $(b \bmod i = 0)$  ТО $c :=$  истина;Конец; ~~начало~~если  $\neg c$  И  $((a \cdot b) \bmod n = 1)$  ТО выведи  $(\text{'Не существует'})$ иначе выведи  $(\text{'Не существует'})$ .

Конец.



Задача 4

Иду по спирале, на первую каскаду пойдет чрепаха, на вторую каскаду - змея, на третью каскаду - лягушка. Найдем участок спиралы, если его ограничивают линиями, а сам участок состоит только из 2-хлемных квадратов. Но волны спирали, длины таких краевых участков будут увеличиваться через один, то есть спирала идёт 2 участка длины  $a$ , затем 2 участка длины  $a+1$  и т.д.

Таким образом, для находящегося на  $k$ -й волне нужно пройти все квадраты, расположенные включим выше участки

Начало

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

Начало

 $S := S + 4$ ;Если  $N = 1$  ТО начало Вывод(4);  $N := 0$ ; конец;Если  $N = 2$  ТО начало Вывод(4);  $N := 0$ ; конец;Если  $N = 3$  ТО начало Вывод(10);  $N := 0$ ; конец;Если  $N > 3$  ТО $S := S + 10$ ;

Начало

 $N := N - 1$ ; \* $S := S + 2$ ;если  $l = 0$  ТО начало  $f := 2$ ;  $K := K + 1$ ;  $i := k$ ;  $S := S + 1$ ; конец;если  $i = 0$  ТО начало  $f := l - 1$ ;  $i := K$ ;  $S := S + 1$ ; конец; $i := i - 1$ ;

конец;

Вывод( $S$ );

конец.

(f)

Задача 5.

Всего даются, мы заполнили или три ~~и~~ одномерных массива, соответствующих стоящим задачам.

Возьмем первый номер и поместим его в строку. Затем возьмем второй номер и поместим ему место в строке, сравнив ~~предыдущим~~ с первым: если второй больше — то разместим его правее, иначе — левее, оставивя пробел между числами. Аналогично будем помещать в строку оставшиеся номера, проходя строку (или строки, если имеется много строк) направо.

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

Задача 6

Пусть школьники пишут по одной или несколько цифр в единице, разделяя числа пробелами (т. е. нечетные единицы, запишиваю их в строку, а при окончании они — переходя на новую). Тогда мы будем аналогично считывать: Н раз по строке из ~~в~~  $M$  символов. Числа будем считывать в строковом формате, собирая в одно наборы между пробелами. Каждый раз, встретив окончание числа (т. е. пробел, до которого нечтного символа), мы будем инициировать переносную единицу так, что всегда будем знать номер пары или результата мы считали. Если это результат — мы присваиваем к общей сумме соответствующего школьника. После всех подсчетов мы определим победителя, сравнив общие суммы школьников.

7

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 70922E 51-80

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ ВАСИЛЬЕВАИМЯ АннаОТЧЕСТВО ОлеговнаДата  
рождения 26.10.1999Класс: 9Предмет ИНФОРМАТИКАЭтап: ЗАКЛЮЧИТЕЛЬНЫЙРабота выполнена на 2 листахДата выполнения работы: 27.02.2015  
(число, месяц, год)

Подпись участника олимпиады:

Васильева

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 изначальной системе счисления цифра, стоящая в определенном разряде не может быть больше  $n$ , где  $n$  это количество деяющих определенных чисел, это максимальная цифра от разряда

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

(+)

Одн. од., двойка

2. Данные надо хранить в таблице:

страна	основное время	Переводят ли часы (Когда переводят (от и до))	Переводят ли часы
--------	----------------	---	-------------------

(x)

В 1 столбце: название страны

Во 2 столбце: время без перевода часов (на przykład, GMT+1)

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

В 4 столбце: период, когда часы переведены (от какого-то часа до какого-то часа),

т. е. время отличается от основного, которое представлено во 2-м столбце

В 5 столбце: в какую сторону переводят время (перед  $+1\text{ час}$ ; назад  $-1\text{ час}$ )

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 столбец):

■ если сегодняшняя датаходит в данный период, то узнать, в какую сторону переведено время (5 столбец):

▲ если +1, то к основному времени прибавить 1, получится исканое время

▲ если -1, то из основного времени (2 столбец) вычесть 1, получится исканое время

■ если сегодняшняя дата не входит в данный период, то время, стоящее во 2-ом

же - исканое.

• если не переводят, то время из 2 столбца - исканое

(+)

3.

•  $R$  присвоить  $R-1$ ;

•  $K$  присвоить 0;

• Выполнить цикл, пока  $R > 0$ : 1)  $T$  присвоить  $R$ ;  $B$  присвоить  $R$ ;

2) Выполнить цикл, пока  $T \neq R$  не равно 0: если  $T > B$ , то

$T$  присвоить  $T - B$ , иначе в присвоить  $B - T$ ;

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

4)  $R$  присвоить  $R-1$ ;

•  $K$  - искомое



9.

· К присвоить 1;

· Тогда на  $M$  раз: К присвоить  $K \cdot N$ ; · К - итоговое число ( $N^M$ )5.  $4 \cdot 11 = 44$ 

· К присвоить 0;

· Выполните цикл, пока  $K : 10000 < 1$ : К присвоить  $K + 44$ ; · Выполните  $K :=$ · Выполните цикл, пока  $K : 1000000000$ : выписать  $K$ ; К присвоить  $K + 44$ .6. Если члено  $S$  будет максимального длины к муру, то компьютер может это сконструировать до мура ( $0+1=1$ ) 

7. К присвоить 44; S присвоить 1;

· Выполните 44 раза: S присвоить  $S \cdot K$ ; К присвоить  $K - 1$ ;

· Т присвоить S; L присвоить 0;

· Выполните цикл, пока  $T > 0$ : Т присвоить целую часть от деления  $T$  на 10 ( $T \text{ div } 10$ );· X присвоить  $L - 1$ ; L присвоить  $L + 1$ ;· X присвоить  $L - 1$ · Выполните цикл  $L - 1$  раз: N присвоить целую часть от деления  $S$  на  $10 \cdot X$  ( $S \text{ div } (10 \cdot X)$ ); если  $N = 4$ , то записать  $X + 1$ ; S присвоить остаток от деления S на  $10 \cdot X$  ( $S \text{ mod } (10 \cdot X)$ ); X присвоить  $X - 1$ ;· Если  $S = 4$ , то записать 1.

Всероссийский олимпиадный лист  
**Олимпиада школьников «Надежда энергетики»**

AN 79-10

← Не заполнять  
Заполняется  
ответственным  
работником

№ группы

Вариант №

7102

шифр

ФАМИЛИЯ Вахрушев

ИМЯ Кирилл

ОТЧЕСТВО Николаевич

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.  $X_8 \rightarrow X_{10} \rightarrow X_{\text{римск. с.}}$

Пример на Pascal:

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

```
begin
```

read(n) // вводимое число в 8-ой с.с.

s:=0; // здесь хранится (будет) десятическая запись числа

n:=n\*10;

j:=0; // счетчик степеней при переводе в десятическую систему

for i:=1 to (n div 10) do // n div 10 - количество цифр в исходном числе

```
begin
```

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

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

s:=s+p^j; // перевод каждого разряда в десят. с.с. и добавление

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

```
end;
```

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

k:=s div 10; // уменьшаем значение на 10

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

for i:=1 to k do write(,x) // выводим все разряды

if s=1 then write(,I); if s=2 then write(,II);

if s=3 then write(,III); if s=7 then write(,VII);

if s=4 then write(,IV); if s=8 then write(,VIII);

if s=5 then write(,V); if s=6 then write(,VI);

```
end.
```



Выборочно последнюю

цифру.

2.

Двоичную-десятичную систему счисления называют, когда мы работаем с двоичной записью числа, а выводим результат в виде десятичной записи числа. Например, если надо перевести число в десятичную:

$21 \rightarrow 12$

$\underline{\underline{0010001}} \rightarrow \underline{\underline{00010010}}$

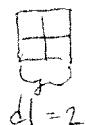


Это двоичное и.к. что несет залог разряда (4 битов) десятичной единицы в двоичном представлении.



## 4. Пример на Паскаль:

$dl$  - длина реки между параллельными берегами шириной  $sk$



$$dl = 2$$

$sk$  - сколько квадратов насыщено

$sr$  - каково загрязнение реки

Var  $dl, sk, sr, n$ : 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

$dl := 2;$

$sk := 4;$

$sr := 12;$

while  $sk < n$  do

begin

$sk := dl \times 4 + 4$



// пока не построим нужное число квадратов  
// квадр. будем насыщать сколько хотим сколько и  
// предыдущее + квадрат на четырех

$sr := dl \times 4 \times 2 + 3 \cdot 4$

// для построения квадратов нужно зря  
// а для симметрии 2/четки

$dl := dl + 2$

// симметрия должна с каждым шагом удваиваться  
// на 2

end;

write(sr); // конец записи реки, вывод

end.

7

5.

Данное будем удобно хранить в массиве с указанием ( $a$ )

$a[i], i$  object - значение параметра  $a[i].result$  - результат  
↑                      ↑  
 $N_{n/n}$               string              longint

Сформировать будем осуществлять - сформированием пузотеркой,



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

Будем считать, что у нас в массиве  $a$ -записи. Все данные могут пропасть сортируются будем брать даты 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+j].result then begin
        z := a[i]
        a[i] = a[i+j]
        a[i+j] = z
      end;
    end
  
```

(7)

Все массивы, т.е. подмассивы, определяющиеся по убыванию, загораются.

6.

Будем считать, что есть стартовая точка  $k$  в без первого массива с  $M$  символами и  $N$  строками ( $a.array[1..M, 1..N]$  от ПАСКАЛЬ)

$K := 1$  // номер символа „№ партии”

(7)

$P := 0$ ; // Обнуляем кол-во очков у Петра и Данила

for i:=1 to  $M \text{ div } 3$  do //  $M \text{ div } 3$  - это как-то таблицу делить на 3.

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

begin  $P := P + a[k+1, j];$  // идем к строке и ее символу (у Петра)

$D := D + a[k+2, j];$  // а дальше уже для Данила

$K := K + 3;$  // меняем указатель на следующий „№ партии”

end;

if  $P < D$  then write („Петр”) else

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

end.

3.↓  
задание



## 3. Пример на Pascal.

```
var n, i, a, b, n1, a1, n1: longint;
begin
  read(n, a);
  k:=1; // Указание начин/перемн
  n1:=n;
  a1:=a; // Создаем дубликаты a и n
  while a1<>n1 do
    if a1>n1 then a1:=a1-n1 else n1:=n1-a1; // находим KDP
  if n1<>1 then write("Не существует") else
    for b:=1 to n do // Найдем b, удовл-
      if (a-b) mod n=0 then begin // берущее условие(доп.к. b<=n)
        write(b); // Выводим подбывающее число
        k:=0; // Указание - налии
      end
    if k=0 then write("Не существует") // если не налии.
  end.
```

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112

0Г61-14

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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.

№4.

бюджет 1 год. макс + 1 год. макс + 1 год. макс + 1 год. макс  
Бюджет

Значит, что 3 складки ровно подавливают все квадраты, которые ставятся сразу после него, как и все дроби в блоке. Итаки складки где  $N$  буда  $k^2+1$  ( $k \in \text{натур.числа}$ ). Там же 3 складки нужно где квадраты, которые бывают где за границу премодульности. Итаки складки где  $k^2+k+1$  ( $k - \text{натур.число}$ ).  $Nst$  - обработка блоком складки. Тогда неизвестное ( $Pascal$ ) будет выглядеть так:

```

if (W=1) then writeln ('4');
if (N > 1) then if (((trunc(N))2+1=N)
or ((trunc(N))2+k+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');

```

Всюду trunc - ф-ия, которая отбрасывает дробную часть, sqrt - ф-ия, которая берет квадрат корень. Использована то, что возводим в квадрат можно отбросить.

№3.

- 1). Скапал описано функцию nod ( $x, y$ ),  
которая вычисляет ~~нод~~  $\text{НОД}(x, y)$  ( $x, y \in \mathbb{N}$   
Эвклида).
- 2) Потом переход от 1 к  $n-1$  можно  
получить  $b$ , чтобы выполнялось  $(a \cdot b) \bmod n = 1$ .
- 3) На калькуляторе (Pascal) напишите ~~логика~~  
алгоритм так:

1). Описание функции nod :

function nod ( $x, y: \text{longint}$ ): longint;

begin

while ( $x <> 0$ ) and ( $y <> 0$ ) do  
if ( $x > y$ ) then  $x := x \bmod y$  else  
 $y := y \bmod x$ ;

nod :=  $x + y$ ;

end;

2). Как алгоритм:

if ( $\text{nod}(a, n) = 1$ ) then for  $i := 1$  to  $n$  do  
if  $(i * a) \bmod n = 1$   
then writeln(i);

if ( $\text{nod}(a, n) <> 1$ ) then writeln('Не существует')



Задание 5.

5). Имеется у нас есть 3 линейных массива:

$a[i]$  - № индекса (строка)

$b[i]$  - значение параметра (столбца)

$c[i]$  - результат (число)

а также  $n$  - количество элементов.

1) ищем число из массива ~~имеющее минимальное значение~~ сорттированного, наименьшее значение "найденное" не исходящего (Pascal):

```
for i:=1 to n do
  for j:=1 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: первым значением должны быть косые  
запятые "предыдущий параметр",  $t$  - переменная,  $w$  - строка



№6. 1) Несколько раз вспоминал забывал. Так что, все скаже какое значение кон. во первое ( $n$ ), а потом  $n$  строкой ему нужно верн.  $k$ -ий, где  $k$ - номер строки, а - кон. бд  
очев Петя, бывшее имя Павел. Короче говоря, в чём его проблема. А также у нас есть 2 лички  $s_1, s_2$  (человек), в которых есть хранилище текущий имена Петя и Павел.

2). На языке Pascal (Pascal) изложи с алгоритмом

также пример текст.

readln ( $a$ );

$s_1 := 0; s_2 := 0;$

for  $i := 1$  to  $n$  do

begin  
readln ( $k, a, b$ );

$s_1 := s_1 + a;$

$s_2 := s_2 + b;$

end;

if ( $s_1 > s_2$ ) then writeln ('Петя');

if ( $s_2 > s_1$ ) then writeln ('Павел');

if ( $s_1 = s_2$ ) then writeln ('Несколько');





№1

Лицо

Аналогичную конструкцию называют напалубко (шахматное поле - N):

- 1). Определение кол-во символов 'M' где веборта, как  $(N \text{ div } 1000)$ , затем  $N_i = N \text{ mod } 1000$ .
- 2) Определение, какими будут первые 100 символов, образом:
  - a). если  $(N \text{ div } 100) = 4$ , то веборты 'CD';  
 $N_i = N \text{ mod } 100$ ;
  - b). если  $(N \text{ div } 100) = 9$ , то веборты 'CM';  
 $N_i = N \text{ mod } 100$ ;
  - c). если  $(N \text{ div } 100) \in [1; 3]$ , то соответствующие концы 'C'
  - d). если  $(N \text{ div } 100) \in [5; 8]$ , то веборты 'D'  
 и  $(N \text{ div } 100 - 5)$  раз веборты 'C'
- 3). Аналогично где разные граничные, то есть:
  - a). если  $(N \text{ div } 100)$  меньше  $(N \text{ div } 10)$
  - b). в пункте: a) веборты 'XL'  
     б) веборты 'XC'  
     в) соотв. концы 'X'  
     г) веборты 'Z' и  $(N \text{ div } 10 - 5)$  раз 'X'
- 4). То же самое где граничные, только:
  - a). если  $(N \text{ div } 100)$  превышает  $N$
  - b). в пункте: a) веборты IV  
     б) веборты 'IX'  
     в) соотв. концы 'I'  
     г) веборты 'V' и  $(N - 5)$  раз 'I'.

На спускной, обратной аналогии на кильватере:



Задание №1 (Чтобчеев):

readln ( $N$ );  
for  $i:=1$  to  $(N \text{ div } 1000)$  do  
    write ('M');  
 $N := N \text{ mod } 1000$ ;  
if  $(N \text{ div } 100=4)$  then write ('CD');  
if  $(N \text{ div } 100=3)$  then write ('CM');  
if  $((N \text{ div } 100) \in [1..3])$  then for  $i:=1$  to  $(N \text{ div } 100)$  do  
    write ('c');  
if  $((N \text{ div } 100) \in [5..8])$  then ~~for~~ begin  
    write ('D');  
    for  $i:=1$  to  $(N \text{ div } 100-5)$  do  
        write ('e');  
    end;  
 $N := N \text{ mod } 100$ ;  
if  $(N \text{ div } 10=4)$  then write ('XL');  
if  $(N \text{ div } 10=3)$  then write ('XC');  
if  $((N \text{ div } 10) \in [1..3])$  then for  $i:=1$  to  $(N \text{ div } 10)$  do  
    write ('X');  
if  $((N \text{ div } 10) \in [5..8])$  then begin  
    write ('L');  
    for  $i:=1$  to  $(N \text{ div } 10-5)$  do  
        write ('x');  
    end;  
 $N := N \text{ mod } 10$ ;  
if  $(N=4)$  then write ('IV');  
if  $(N=3)$  then write ('IX');  
if  $(N \in [1..3])$  then for  $i:=1$  to  $N$  do  
    write ('I');  
if  $(N \in [5..8])$  then ~~for~~ begin  
    write ('V');  
    for  $i:=1$  to  $(N-5)$  do  
        write ('I');  
    end;

№2.

Ломан

- 1) Задана свободное и следующее вопросы:  
нужно настроить такое число  $N$ , чтобы при делении  
второго хранить в 2-10 остатка деления,  
чем в 2.
- 2) а). Стартуя  $k$ -максимальным членом числа  $N$ ,  
когда делится это в сущ. врем. 2 раза будет  
закончено ЧК без
- б). Стартуя  $x$ -максимальным возможным числом,  
когда  $2^x \leq N$ , когда в 2 способа деления  
будут закончены  $(x+1)$  без.
- 3). Рассматривается настройка такого  $N$ , чтобы  
 $ЧК \leq (x+1)$ .
- 4). Рассматриваются  ~~$k+1$  для~~ ручные  $k$  и при  
них ~~наименьшем~~ максимальном  $x$ :
- $k=1 \Rightarrow \max x = 3 \Rightarrow 4 < 4$  (неверно)
- $k=2 \Rightarrow \max x = 6 \Rightarrow 8 < 7$  (неверно)
- $k=3 \Rightarrow \max x = 9 \Rightarrow 12 < 10$  (неверно) ?
- $k=4 \Rightarrow \max x = 13 \Rightarrow 16 < 14$  (неверно)
- $k=5 \Rightarrow \max x = 16 \Rightarrow 20 < 17$  (неверно)
- $k=6 \Rightarrow \max x = 19 \Rightarrow 24 < 20$  (неверно)
- Однако это и при добавлении к ручной настройке  
 $x+1$  и ЧК будет только разница. Следовательно,  
2-10 остатка деления не будет ~~закончено~~ но только  
закончено, но не для разных чисел. Единственное  
это число которое в том, что ~~закончено~~ с ней кроме про-  
изводства операции перевода в гр. деления деления.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7113

SG81-88

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ Гончаренко

ИМЯ Дмитрий

ОТЧЕСТВО Александр

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.



Dmitriy  
genn  
it  
in

□

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

кн}

Введем массив  $u$  из  $p$  элементов  $\text{int } u[p];$ Заполним этот массив нулями  $V[1..p] = \{0, 0, 0, \dots, 0\}$ Присвоим  $k$  значение  $p-1$  ( $k=p-1$ );Пока  $k \geq 0$ иначе  $\{ n = n - V[i];$  // присвоим  $k$  разность  $n$  и ближайшего

меньшего числа Фибоначчи

 $u[i] = 1$ ; //  $i$ -<sup>ый</sup> элементу массива и присвоим единицу,

что соответствует позиции числа Фибоначчи, которое

мы вышли в предыдущем действии.

Если  $n$  не равнается 0, то иначеПока  $(V[k] > n) \{ -k; \}$ Иначе  $i = -1;$  // выйден из цикла

кн}

иначе  $Borlegim$  массив  $u$ , начиная с последнего элемента

(получим число Фибоначчи)



4

Ввод  $M \times N$ .По ка  $M > 0$ 

$$\text{кд} \quad \begin{array}{l} N = N * N; \\ M = M - 1; \end{array}$$

(—)

кд

Вывод  $N$ ;

5

Ввод  $M \times N$ ;Создадим массив  $V [M \times N]$  (сам строка)Цикл кот 0 go  $M \times N$ 

// for (int k=0; k&lt;M\*N, ++k)

кд Циклом k go  $M \times N$ 

for (int i=k, i&lt;M\*N, ++i)

кд // если ( $V[i] < V[k]$ ), то меняем их местами(swap ( $V[i], V[k]$ ))

кд

кд

Цикл кот 0 go  $M \times N$  кд // строкаЦикл i от 0 go  $M$  // строкикд // Выводим значение массива  $V$  (++M)

кд // переходы на следующий столбец

кд // Переходим на следующую строку (++N)

кд // Переводим на следующую строку (++N)

кд // В результате получим таблицу  $M \times N$  заполненную убывающим

знач.

(+)



3

Blog:  $\text{int } p \geq 2; k=p-1, s=0; t=\log_2(p)$ .  
 bool test=true.

~~for~~ ( $k : \text{на 2}$ )

и<sup>и</sup>  $K$  делит на 2

( $k=k/2;$ )

s увеличиваем на единицу (++s;)

к<sup>к</sup>

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

$$k=t;$$

/\* Продолж \*/ Пока  $k > 0$

и<sup>и</sup> Выберем случайное число  $a$  от 1 до  $p-1$

Если  $a^d$  не равняется 1 по модулю  $p$ , то  $\text{test}=\text{false}$ .

--k; (уменьшаем  $k$  на единицу)

к<sup>к</sup>

/\* Продолж \*/ Цикл от  $r=0$  до  $s-1$

и<sup>и</sup> Пока  $t > 0$

и<sup>и</sup> Выберем случайное число  $a$  от 1 до  $p-1$

Выберем цел  $m=d/2;$

Если  $a^m$  не равняется -1 по модулю  $p$ , то  $\text{test}=\text{false}$

--t; (уменьшаем  $t$  на единицу)

к<sup>к</sup>

Если  $\text{test} == \text{false}$ , то  $\text{Blog}$  "составное"

Иначе  $\text{Blog}$  "верастно простое".





W G

6 **Диаграмма зон загара на концепцията чукът.**

Со зваженням чинок стиків. (Т.к. ніде нічого не видає)

*Corygallum* чисток чорногуб

Документ является в список отходов генерации его транспорта.

(первый элемент списка строк  $\text{array}$  содержит значение  
первой строке в которой есть команда  $\text{By}$  (если есть)  
(т. к. расположение сменных неизвесто, будем искать ее, т.к. другое у)  
затем мы будем ходить на  $x$  вправо на  $2m^2$ )  
 $k=0 \quad i=1$

(one firm's optimal payment depends on  $S = 1000k * 1000L$ ,  $k=0, i=1, n=1$ ;  $\pi_i = \pi_1$ ; or  $\downarrow$ )

Пока в строке есть элементы  
из списка в строке ~~строки~~ добавляем элемент и сдвигаем его

текущий  
Запуск вкл-го спускает y smai em ji;  $P = P - 1$ ;  
 $S = S - 2$ ;  
 $++j;$

-Установим текущим элементом списка строк первого элемента в строке. Установим текущим элементом списка строка стартовая.  $n=1$ ;

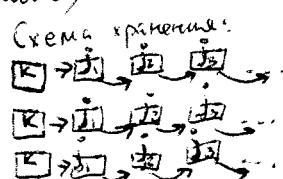
Перейдем на первый элемент списка строк из.

Создадим массив  $V[N]$  элементов. и заполним его нулями.  
 Пока существует неизвестный элемент списка строк. мы  
 Пока в списке есть текущий элемент списка строк  $k > 0$

ky

(Получил кон-то even когда bug )  
самоустановка не входит)

Бибогум  $P_p - P$  (скоро) заменяет не биомасу)



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7102

*СШ 59 - 96*

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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:

*Григорьев*

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. *Будет 1 др. лист.* *Будет 2<sup>м</sup> др. лист* *Будет 3 др. лист*  
*Будет 4 др. лист №4* *Будет 5 др. лист*  
*Будет 6 др. лист №5*

В этой задаче я нашел закономерность:

1) 4 ряда потребуется только в одном случае *Будет 1 др. лист №4*  
для одного квадрата.

2) 3 ряда потребуется в том случае если  
 $(n-1)$ -ий квадрат целого числа.

$(n-1)$  равно (целая часть корня из  $n$ ) уменьшенному на (целую часть корня из  $n+1$ ).

3) 2 ряда во всех остальных случаях.

Решение к этой задаче я сделал на  
языке Free Pascal.

7

```
program z4;
var n: integer; i,s: longint;
begin
readln(n);
if (n=1) then writeln ("Потребуется 4 ряда")
else
begin
s:=4;
for i:=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 ("Потребуется", " ", s, " рядов");
end;
end.
```

Получаем, что в  $s$  всегда находится минимальное  
количество рядов.



N 1

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

program z1;

var

function iz8v10 (n:longint): longint;  
var i1,j1,step,ch,ans,x:longint;  
s: string;  
begin  
str(n,s); {переводим число в строку s}  
ans:=0;  
for i1:=0 to length(s)-1 do  
begin  
step:=1;  
for j1:=1 to i1 do  
step:=step\*8;  
~~val(s[i1+1]~~  
val(s[length(s)-i1], ch, x);  
ans:=ans+ch\*step;  
end;  
iz8v10:=ans;  
end;

function izRim(n:longint): string;  
vars: 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 + ' ' 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';
                end
        end
    else
        S := S + 'I';

iz Rm := S;
end;
```



Богдан - 1 год, учр.

Югор

begin  
Readln(n);

if  $n \leq 0$  then writeln("Невозможно  
перевести") else

begin  
writeln(izRim(iz8v10(n)));  
end;  
end.

N 3

Составить программу на языке Free Pascal:

program z3;  
var n, d, b, i: longint;  
function NOD (n, m: longint): longint;  
var min, i, x: longint;  
begin  
if n > m then min := m else min := n;  
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, d);

реш. лист №2  
БГК

if  $\text{NOD}(a, n) < > 1$  then writeln("Не существует") else

begin

$b := 0;$

while  $(a - b) \bmod n < > 1$  do

inc(b),

writeln(b);

end;

end.



№ 6.

В условии сказано как расположены входные данные, поэтому я буду считать, что в одной строке более трех чисел (1 партии, pes 1, pes 2).

Сказано, что размер страницы  $M \times N$ , значит всего было сорвано  $M$  партий.

Решение написано на языке Free Pascal:

program Z6:

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

begin

Readln(M, N);

Petr := 0; PAN := 0;

for i := 1 to N do

begin

readln(p, rez1, rez2);

Petr := Petr + rez1;

PAN := PAN + rez2;

end;





begin var, such №3  
Ясенев

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

N5

Всё я буду проводить построение, т.е. в одну строку вводу и Nп/п, значение параметра и результат.

хранить это буду в 2 массивах. один массив - мас-  
сив строк, в нем будет хранится Nп/п и значение парамет-  
ра, в другом результат;

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

Писать буду на Free Pascal:

```
program z5;
var n, xi, j: longint; s2, 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];
    while i < j do begin
      while mas[i] > X do i := i + 1;
      while mas[j] < X do j := j - 1;
      if i <= j then begin
        temp := mas[i];
        mas[i] := mas[j];
        mas[j] := temp;
        i := i + 1;
        j := j - 1;
      end;
    end;
    quicksort (l, i-1);
    quicksort (j+1, r);
  end;
end;
```

7



богдан жон-маком №  
Лсенор

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] := mas1[i];  
mas1[j] := temp; mas1[i] := temp1;  
inc(i); dec(j);  
end;

Until i < j;

quicksort (i, p);

quicksort (l, j);

end;

begin

n := 0;

While not eof do { пока не конец файла }

begin

Readln(s);

s1 := " "; s2 := " ";

j := length(s);

n := n + 1;

while s[j] <= ' ' do

begin

s1 := s[j] + s1;

dec(j);

end;

for i := 1 to j - 1 do

s2 := s2 + s[i];

mas[n] := s2;

val m (st, k, c);

{ символ = обозначает  
пробел }



Бурак Дж. Истор № 5  
Узбекистан

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

Вывод можно убрать т.к. в условии задачи не сказано выводить результат или нет. А алгоритм сортировки, который я применял является одним из самых быстрых алгоритмов сортировки из всех существующих. Он работает за  $\log_2(n)$ .

Поэтому сортировать программы будет быстро. Однако вводнее по памяти использовать двоичный формат для хранения данных, но это не обязательно, но так как с ним работать.

N 2.

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

Это удобно тем, что перевод в двоичную систему счисления легко осуществляется с помощью триад и метров. Или же выполнить операции.

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

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7102  
*СЧ 59 - 38*

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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:

*Исаев*

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 ~~Будет 1 год. Исполнилось 20 лет с дн. исп.~~  
~~Будет 1 год. Исполнилось 20 лет с дн. исп.~~

~~Решение~~

Пусть  $k$ -значное число в восемнадцати системе счисления. ~~( $k$ -значное, т.к. в противоположном случае было бы не  $k+1$  разрядов)~~  
 Мы можем перевести его в десятичную систему счисления.  
 Для этого находим каждый разряд числа записанной суммой:  $K = \overline{abc...n}_8$  и проконвертируем каждый разряд в шестнадцатеричную систему счисления и отразим наше значение  $n$ .

$K = \overline{\cancel{abc...n}}_8$  (~~если~~  $n$  кратно 8, скажу ~~левое~~ ~~левое~~ ~~левое~~ ~~левое~~ ~~левое~~ ~~левое~~)

$$\overline{\cancel{abc...n}}_8 = 8^f \cdot n + \dots + 8^1 \cdot c + 8^0 \cdot b + 8^{-1} \cdot a_{10} = m_{10}$$

Таким образом, мы получаем  $k$ -значное число  $K$  в десятичной системе счисления и отразим его как  $m$ .

Теперь переведем  $m$  в римскую систему счисления.

Пусть  $m_1$  — ~~остаток~~ частное от деления  $m$  на 1000. В римской

системе счисления число будет иметь  $m_1$  знаков  $M$ .

Пусть  $m_2 = \frac{m}{1000}$  остаток от деления  $m$  на 1000. ~~частное от деления из~~

~~частное~~

$d_3$  — остаток от деления  $m_2$  на 10.

~~частное от деления~~

$$d_2 = (m_2 \text{ div } 10) \text{ mod } 10 \quad (\text{div} - \text{частное от деления}; \text{mod} - \text{остаток от деления})$$

$$d_1 = m_2 \text{ div } 100.$$

Если  $d_1 < 4$ , то новое число  $x_2$  будет состоять из  $d_1$  знаков  $C$ .

Если  $d_1 = 5$ , то  $x_2$  будет состоять из знаков  $D$ .

Если  $d_1 \in [6; 8]$ , то  $x_2$  будет состоять из числа  $D$  и остатка  $(d_1 - 5)$  знаков  $C$  в соответствующем порядке  
 $(x_2 = D \underbrace{C \dots C}_{d_1 - 5})$

Если  $d_1 = 9$ , то  $x_2$  будет состоять из знаков  $C$  и  $M$  ( $x_2 = CM$ )



~~Доказательство на основе векторов~~

Если  $d_2=9$  и  $d_3=9$ , но  $d_1 \neq 9$ , то  $X_2 = IM$

Если  $d_2=9$  и  $d_3=9$ , но  $d_1 \neq 9$ , но  $d_4 \geq 4$ , то

~~X2 = D C..C IC~~

$$X_2 = \underbrace{D}_{d_1-5} \underbrace{C..C}_{d_2-5} IC$$

Если  $d_2=9$  и  $d_3=9$  и  $d_4 \leq 4$ , то  $X_2 = ID$ .

Если  $d_2=9$  и  $d_3=9$  и  $d_4 < 4$ , то  $X_2 = \cancel{I..C/C} \underbrace{C..C}_{d_1} IC$ .

~~также~~  $\Rightarrow 9$

Для всех остальных случаев ~~X2 записывается~~  $X_2$  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. = ~~II~~  $\underbrace{C..C}_{d_1}$

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

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

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  $(d_4 > 4) \cup (d_4 \leq 3)$ , то  $d_4$  в рим. с.с. =  $\cancel{DC..C}$

Если  $(d_2 > 4) \cup (d_2 \leq 3)$ , то  $d_2$  в рим. с.с. =  $\cancel{D..C}$

Если  $(d_3 > 4) \cup (d_3 \leq 3)$ , то  $d_3$  в рим. с.с. =  $\cancel{X..X}$

Если  $d_1=3$ , то  $d_1$  в рим. с.с. =  $CM$

Если  $d_2=3$ , то  $d_2$  в рим. с.с. =  $XG$

Если  $d_3=3$ , то  $d_3$  в рим. с.с. =  $IX$

$$X_2 = \overline{d_1 d_2 d_3}_{\text{рим. с.с.}}$$

( $X_2$  состоит из последовательных чисел  $d_1, d_2$  и  $d_3$  в римской с.с.).

Получаем, что наше число  $m$  в римской системе счисления равна  $\overline{X_1 X_2}$  ( ~~$X_1$  и  $X_2$  записываются последовательно~~).



Помоги образуми, как перевести восьмеричное число к восьмидесяти шестнадцати системам счисления.

Задача №4							
26	27+	28+	29+	30+	31		
+							
25	10	11+	12+	13	32+		
				+	33+		
24+	9+	2	3	14			
23+	8+	1	4+	15	34+		
22+	7	6+	5	16+	35+		
21	20+	19+	18	17	36+		

## Решение

Признаком отыскания в таблице все квадраты, для построения которых необходимо 2 ряда.

Мы видим, что все квадраты, кроме которых — чётные числа,

крайне меч, которые имеют вид  $n$  (которые квадраты) =  $= i \cdot i + 1$ , строятся с помощью двух рядов.

Это происходит оттого, что все квадраты вид  $n = i \cdot i + 1$  выражаются из присоединяющихся к квадрату  $i \cdot i$ , то есть образуют узелок. Все квадраты, образующие узелок строятся с помощью 3 рядов. А все квадраты имеющие чётные номера, но не имеющие вид  $n = i \cdot i + 1$  строятся уже к узелку и дополняются до него нового узелка если до прямоугольника, потому что их построение необходимо 2 ряда.

Строй Все квадраты с чётными номерами и видом  $n = i \cdot (i-1) + 1$  строятся из присоединяющихся к узелку  $i \cdot (i-1)$  и дополняются до ~~2~~ узелка, т.е. где они необходимы 3 ряда. Все основные чётные квадраты строятся до ~~2~~ узелка, потому что где они нужны 2 ряда. (Изложение из правил 1-ой задачи, к четверти-Ч ряда).

Напишем программу на языке Паскаль:

```
var n,i,s: longint;
begin
readln(n);
if n mod 2=0 then
begin
s:=0; {s-счётчик}
```





```

for i:=1 to n do
if (i*i+1=n) then s:=s+1;
if s=0 then writeln (2) else writeln (3);
end;
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:=-1;
  for i:=1 to n do
    if (a*i) mod n=1 and (prost(a,n)=1) then b:=i;
  if b=-1 then writeln ('Не существует') else writeln (b);
end.

```

{ в функции: если общий делитель 1, то число взаимопростое, если нет, то не взаимопростое }

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



зап. лист №2  
Лар

## Задача 5.

Решение.

Решим эту задачу на языке Pascal. Нам даны 3 одномерных массива  $a$ ,  $b$  и  $c$  и  $n$ -число экспрессионов. С помощью сортировки пузырьком все данные в массиве с небольшим количеством ошибок в соответствующем образе изменят 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 b[c[j]] > b[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;
        for i := 1 to n do
          writeln(a[i], b[i], c[i]);
        writeln;
      end;
end.
```

⑦

## Задача 2.

Решение

⑦

Найдите четырёхразрядное десятичное число которое можно записать в двоично-десятичной С.С. (Четырехбитное ~~четырёхбитное~~ двоичное коде десятичного числа от 0<sub>2</sub> до 111<sub>2</sub>, т.е. от 0 до 15). Значит найдите четырёхразрядное десятичное число которое можно записать в двоич-десим. С.С., кроме определенных чисел, т.к. не является разрядов для  $-$  ~~один~~  $\rightarrow$  ~~все остальные~~  $\rightarrow$  ~~один~~



Наиболее возможен и вероятный, когда имеется <sup>богатое пер. муз. № 3</sup> паспортное значение числа непропорциональной дисгармонии <sup>или</sup> **ИСХОД**.  
В фортепиано-гитарной С. С. как выражено звучание <sup>внешнее</sup> числа <sup>внутреннее</sup> звуков звуков когда восстановлены звуки от 0 до 15. Значим  
в фортепиано-гитарной С. С. все возможные гармонии все возможные непропорциональные числа в аксессусах звучания не более 16-ти звуков.  
N 6. Омблем: Все пас. наст. числа в системе  
Решение <sup>личн.</sup> не более 16-ти звуков.

Где  $\gamma$  — гана симметрия  $M \times N$  кубиков. Ганане прозрачны на языке Pascal:

program ZI

~~Var a: array [1..1000, 1..1000] of longint;~~

m, S1, S2, n, i, j: longint;

begin

readIn (m, n):

for i := 1 to m do

for j := 1 to n do

Readln (aci, r3);

{ допустим, что ~~наиболее~~ <sup>нельзя</sup> ~~посчитать~~ <sup>суммой</sup> свои результаты в ~~счет~~ <sup>суммой</sup>; все ~~составов~~, ~~наиболее~~ <sup>составов</sup> при делении на 3 ~~разом~~ <sup>остаток</sup> останется ~~наиболее~~ <sup>наиболее</sup> посвящены ~~наиболее~~ <sup>наиболее</sup> ~~стабильных~~ <sup>стабильных</sup> комбинаций при делении на 3 ~~разом~~ <sup>остаток</sup> 2 посвящен результатам Пети, а ~~наиболее~~ <sup>наиболее</sup> стабильных комбинаций при делении на 3 ~~разом~~ <sup>остаток</sup> 0 посвящены результатами Дашкин } }

$S := 0;$       {  $S$  - одисмі поз-т Нема,  $S_1$  - одисмі поз-м Коли }  
 $S1 := 0;$

for i := 1 to m do

for j := 1 to n do

if ( $i \bmod 3 = 1$ ) then  $S := S + a[i, j]$ ;

if ( $\max B = 0$ ) then  $S1 := S1 + a[i, \tau];$

End

If  $s > s_1$  then write in ('long');

If  $S = S_1$  then written ~~( $S_1$ )~~  $\downarrow$  (‘nurs’);

if  $S \leq S_1$  then write  $n$  ('Memo');

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7103

04 30 - 20

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ

ИсрАПИЛОв

ИМЯ

МАХАЧ

ОТЧЕСТВО

ИсрАПИЛОвич

Дата

рождения

05.01.1990

Класс:

10

Предмет

Информатика

Этап:

ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на

2

листах

Дата выполнения работы:

15.03.2015г.

(число, месяц, год)

Подпись участника олимпиады:

И.А.И

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



Найдите такие два последовательно идущих в ряду Роджерса числа  $x_n$  и  $x_{n-1}$ ,  $x_{n-1} < x_n$ , для которых выполняется условие  $x_{n-1} \leq x \leq x_n$ , где  $x$  - число, которое мы можем представить в двоичной системе счисления. Из  $x$  вычтите  $x_{n-1}$  и умножьте это  $x_{n-1}$  на свое выражение  $x - x_{n-1} = x_1$ . Ответом числа  $x_1 \geq 0$ , то мы получим двоичное значение, при котором  $x = x_1$ , где здесь  $x_1$  не содержит единиц. Потом определите  $x_1$  и запишите первое, где первого  $x_1$ . Наши значения, соответствующие  $x_{n-1}$ , замените единицами, на единицы-нули. Представив полученные последовательности нулей и единиц в обратном порядке, получим итоговое представление означенного числа в двоичной системе счисления.

№2



Запишем предположение на проверку числа как  $p$ . Для уравнения  $p-1 = 2^k \cdot d$  найдем  $s$  и  $d$ . Будем делить  $p-1$  на 2, пока не поделим на чётное число. Это и будем  $d$ , а остаток деления —  $s$ . Введём  $t = \log_2 p$ . Тогда  $t$  — это  $k$  единиц в разрядной записи  $p$ . Теперь  $t$  раз случайно выберем  $a$  из  $\{1, 2, \dots, p-3\}$ .



1)  $a^d \equiv 1 \pmod{p}$ .

2) Было бы  $t \mid 0 \leq s \leq 1$ , где  $a^t \equiv -1 \pmod{p}$ .

Для проверки первого условия введём  $a^t$  и проверим равенство  $a^t \equiv -1 \pmod{p}$ . Для проверки второго условия это будет  $t \in [0, 3]$ , и для каждого случая проверим равенство  $a^t \equiv -1 \pmod{p}$ .

Если для всех  $a$  сгенерированных случайных страведибо, то, вероятно, число простое, в противном случае это число составное.

№3

Найдите представление в виде суммы дружеских чисел. Число представимо и в виде суммы четырёх чисел, или трёх, или.

$$N^{11} = N^{(m_1+m_2+\dots+m_k)} = N^{m_1} \cdot N^{m_2} \cdots N^{m_k}$$

При  $m_1$  генераторно меньше ( $m_1 \leq 3$ ), выражение представляющее в виде простого членом. Теперь необходимо представить  $M$  в виде ряда чисел  $m$ . Для этого ~~заполните~~ будем учитывать  $M$  числа, которые и есть в последовательности из  $1$  начиная с третьего. Первые 3 числа должны быть из чисел — простых чисел. Каждое из них должно быть представлением в виде суммы

N2 - нет  
N3 - нет



Повторяющаяся, члены из  $2 \times 2$  матрицы, складываются в матрицу размером  $N \times N$ .  
 Повторяющиеся операнды из матрицы складываются в матрицу размером  $N^M \times N^M$ .  
 $N^M = N^{m_1} \times N^{m_2} \times \dots \times N^{m_k}$ , называемую  $N^M$ .

13

Биотопът на рептилите е също така разнообразен и съвсем не съществува един  
единствен биотоп за всички видове. Така например в м. Македония са  
представени и горски и степни, и пустински, и саванни, и тундрови, и  
пещерни, и подводни биотопи. Всички те са свързани със степните биотопи.

C.3

Геном - гетерогенный набор генов, в котором есть как регуляторы (активаторы и трансформаторы), так и структурные гены, которые не кодируют белка. Достаточно много в геноме нет генов, которые кодируют белки, т.к. есть гены, которые кодируют белки только в определенных условиях.

6

Две различные задачи возбуждения ячейки  $I \cdot R$  на  $L \cdot 1000$  разного по  $R$   
Часть I. Каскадное разд. возбуждение на  $R = 1000$  ом. Каскадные ячейки на  $N$  элементах, предшествующему разд. Каскадный разд. – элементы стоят в один ряд. Каждая ячейка разд., имеющая общий разд. – элементы стоят в один ряд. В  
 каскадной структуре максимальный сигнал есть. В элементах стоят, соединяющихся с общим, суммарный напряжением, соединяющим каскады группами, между которыми разделяются. Если общее напряжение, запоминающееся в ячейке  $O$ , то изображение на  
 запись неоднократно делается в ячейке. Сформированное еще один разд., (с  
 общим элементом)  $M-N+1$ .  $M-N+1$  раз разделяется по первому стояку (один-  
 му, единому), но разд. группируется в ячейку 2. Каскадный разд. будем  
 предполагать равнозначимый звуком  $I_{N+1}$  в 1 ячейке, который записан звук, со-  
 храняя который записан в ячейке  $2+N-1$ ; и запись повторяется вновь в ячейке 2, и  
 так же записывается ячейка 2. Таким образом, в результативные стояки 2 сумми-  
 руются соответствующие звуки разд. ячейки  $M-N+1$  в конечном итоге –  
 получим разд. ячейки  $M-1$  до  $M-N+1$  соответствующий.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

3093

УК 48-25

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

**ФАМИЛИЯ** КОВАЛЕВ

**ИМЯ** Даниил

**ОТЧЕСТВО** ЮРЬЕВИЧ

**Дата рождения** 25.04.2001

**Предмет** Информатика

**Работа выполнена на** 4 **листах**

**Класс:** 9

**Этап:** Заключительный

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

Подпись участника олимпиады:

Д.Куя

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





№2

В ЭВМ используется двоичная С.С. из-за своей простоты, т.к. в ней всего 2 цифры - 0 и 1. Информация передается с помощью электрических сигналов, и 0 соответствует отсутствию сигнала, 1 - сигналу. Если бы для использования двоичной системы, как когда были только 10 уровней десятичной С.С., то с угодом не могло бы времена первых ЭВМ сократить невозможно. Тогда сейчас есть технические возможности сделать ЭВМ, работающую в десятичной С.С., однако "на грядущем" уже придется использовать двоичную, т.к. это стала бы причина. Используя двоичную С.С. - числа в ней очень удобные, забавные и интересные и воспринимают их системы счисления, кроме десятичного. Они являются "произведениями" от двоичной, и перевод из одной такой системы в другую очень простой и быстрый. Такие системы используют для шифрования чисел знаков в числе, например,  $16_{10} = 10000_2 = \text{F}_{16} 10_{16}$ . Например, адреса памяти в компьютере хранятся в шестнадцатеричном формате.

(+)

№1

Римская система счисления называется позиционной. Действительно, ведь цифра 0 или 1 имеет значение разное в зависимости от того, на какой позиции находится. Например, в числе  $100^1_{\text{fib}}$  первое 1 значит 5, второе 1, третье. Исходя из этого, римская система счисления:

(+)

доказательство:



№7

В первом десятке таких чисел первое есть: 2, 3, 5, 7. В других же десятках простыми могут быть лишь нечетные числа, т.е. кончивающиеся на 1, 3, 5, 7, 9, но число, кончивающееся на 5, делится на 5, и это составное (также оно делит любое число на 5, но это распространено с любым десятком). Таким образом, будем проверять из каждого десятка 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 = 1; i <= 4; i++)
        if (a == i)
            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 набираем простые числа проверки вручную, также сразу проверим делительность на 2 и на 3. Далее стандартно будем перебирать все числа до  $\sqrt{a}$ , если какое-то из них делится  $a$ , то оно составное. Число — простое. Это реализовано в функции `Prost()`; вычисление четвёрки в виде на экран — в функции `main()`.





№5

$$\begin{array}{r}
 \bullet 1 \\
 \times 3 \cdot 2 \\
 \hline
 830 \\
 + 3 \cdot 2 \\
 \hline
 1 \cdot 8 \cdot 3 \cdot
 \end{array}$$

Понятно, что во втором множителе последняя цифра 0 (поскольку это  $x$ , имеем  $3+x=3$  или  $4+x=3$  (если были переходы через разряды)), отсюда  $x=0$  или  $x=3$  (в случае перехода через разряд), но перехода быть не может, следовательно, это 0). Первая цифра ~~первого~~ множителя - это 1 (в сумме только эту цифру получается 1).

При умножении  $3 \cdot *1* = 12 \cdot 5$ , отсюда последняя цифра первого множителя - это 5, переходом подтверждаем что первая цифра - это 4, т.к. же подтверждение 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:  $x+4+1=8$ ,  $x=3$ , отсюда вторая цифра второго множителя  $3320 : 415 = 8$ , и весь результат вычислить с обратной целью обозначим:

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

9

№6

Переводим байты по стандартному алгоритму, используя цифры на значение соответствующего разряда, например,  $132_4 = 1 \cdot 4^2 + 3 \cdot 4^1 + 2 \cdot 4^0 = 16 + 12 + 2 = 30$ . Представим число байт первого в числовом формате (выясним это самим сейчас), используя старшую цифру переведём в соответствующие числа ( $A \in 10$ ,  $B \in 11$ ,  $C \in 12$ , ...,  $F \in 15$ ). Полученное в конце число приведём к строковому типу. Результативное же C++:



## Олимпиада школьников «Надежда энергетики»

Вариант: 7093

ШИФР, НЕ ЗАПОЛНЯТЬ! ⇒

УК 48-25

```

#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 такие же
    // операции с цифрами от
    // 4 до E
    else if (c == 'F')
        return 15;
    else
        return -1; // ошибка при вводе числа
}
int power(int a, int b) // возведение 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); // System считает, что одинаковы при
                                                               // вводе числа нет, иначе просто
    string s = (string)ans;                                         // добавим оператор if (toNumber(
                                                               // [kx[i]] != -1)
    cout << s;
    return 0;
}

```

№3

```

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

```



Находим  $d$  из  $s$ . Для этого нужно сделать  $p-1$  на 2, пока это еще делится, и один делится, убирая вправо 1 на 1. Ког да C++ для этого умножка представил ненее. При проверке условие  $a^d \equiv 1 \pmod p$  будет выполняться при умножении на 2 браз остаток от деления на  $p$ , что избывает квадратичный, и получившийся величина должна быть второй строке 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.

Одномерный алгоритм на языке Pascal:

Var x, pp, p, n, r, ch: integer; {x - введенное число, pp, p, n - числа  
цифровой записи, r - для сдвига разрядов, ch - результат из

Begin

```

read(x);
if (x<0) or (x>abs(x)) then
  writeln('некорректные данные')
else
  begin
    pp:=1; {начало последовательности Раджаками}
    p:=1; {для нашего алгоритма нам нужно знать}
    n:=2; {этих последовательных чисел Раджаками}
    while (n<=x) do {этим циклом мы находим
      begin
        pp:=p;
        p:=n;
        n:=p+pp;
      end;
      n:=p;
      p:=pp;
      pp:=n-p;
      r:=1;
    }
    if (x<=n) then {делаем так, чтобы макс. разряд
      begin
        ch:=ch-r+1;
        r:=r*10;
        x:=x-n;
      end;
    }
    if (x>n) then {если число не входит в раз-
      begin
        ch:=ch*r;
        r:=r*10;
      end;
    }
    n:=p;
    p:=pp;
    pp:=n-p;
  end;
  ch:=(ch div 10);
  writeln(ch);
end;

```

End.

Схема алгоритма:

- 1. Вводим число x.
- 2. Проверяем, что x - целое и неотрицательное.
- 3. Инициализация переменных: pp = 1, p = 1, n = 2.
- 4. Цикл while (n <= x):
  - 4.1. Вычисление pp := p, p := n, n := p + pp.
  - 4.2. Установка n := p, p := pp, pp := n - p.
  - 4.3. Установка r := 1.
- 5. Если x <= n, то:
  - 5.1. Вычисление ch := ch - r + 1, r := r \* 10, x := x - n.
- 6. Если x > n, то:
  - 6.1. Вычисление ch := ch \* r, r := r \* 10.
- 7. Установка n := p, p := pp, pp := n - p.
- 8. Вычисление ch := (ch div 10).
- 9. Вывод результата ch.



Задача №5.

```

Var a: array [1..1000] of integer; { массив для чисел }
m, n, i, j: integer; { m, n - размер массивов, i, j - переменные для циклов }

Begin
    read(m, n); { считываем размеры }
    if (m <= 0) or (n <= 0) then
        writeln('данные некорректные'); { проверяем данные на правильность }
    else
        Begin
            for i := 1 to m do
                for j := 1 to n do
                    read(a);
            for i := 1 to m * n do { считываем дальше в одно -
                read(a[i]); } { мерный массив }
            for i := 1 to m * n do
                Begin
                    for j := 1 to m * n do
                        begin
                            if (a[j] < a[j + 1]) then
                                begin
                                    d := a[j];
                                    a[j] := a[j + 1];
                                    a[j + 1] := d;
                                end;
                            end;
                        end;
                    for i := 1 to m * n do
                        write(a[i], ' ');
                end;
            end;
        end;
    end;

```

Задача №3.

```

Var b: array [1..1000] of boolean; { массив для результатов проверки }
p, d, s, t, i, pr, pr2, pb, a, j, l: integer; { l: результат; 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 do
                while (d mod 2 = 0) and (d div 2 = d / 2) do { проверка деления на 2 и находят d и s }
                    begin
                        d := d div 2;
                        s := s + 1;
                    end;
            l := s;
        end;

```

проверка на следующем месте...



## Задача №3

```

t := log2(p);
for i := 1 to t do { в этом цикле проверяется, что делит p на α }
begin pr := 1; pr2 := 1; pr3 := 1; flag := false;
α := randomize(p - 2) + 1; { находят числ. α из диапазона }
for j := 1 to d do { в этом цикле находят α в степени d }
pr := pr * α;
for j := 0 to s - 1 do { в этом цикле проверяется все r }
begin
for l := 1 to j do { в этом цикле находят k }
pr2 := pr2 * 2;
for l := 1 to r { в этом цикле находят α в степени k - d }
pr3 := pr3 * α;
if (pr mod p = 1) or (pr3 mod p = p - 1) then
flag := true; { проверка выполняется в условии }
end;
α[i] := flag;
end;
flag := true;
for i := 1 to t do
if (α[i] = false) then
flag := false;
if (flag := true) then
writeln('вероятно простое') { вывод результатов }
else
writeln('составное');
end;
end.

```

{ записываем результаты проверки в массив для каждого α }

{ если для хотя бы 1 α проверка не проходит, то мы делаем вывод, что число составное }

## Задача №4.

```

Var a, b: array[1..1000] of integer; { в-массив для цифр N α- }
m, n, i, kn, p, r, j, l, ps: integer; { в-перем. для цифр 'alpha' }
{ в-перем. в след. разряд, ps-перем. в
Begin
read(m, n); { считывание данных }
while (n <= 0) do { записываем цифры числа n
begin
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; { ps, p - перемен. в след. разряд }
ps := 0; { ps, p - перемен. в след. разряд }
{ продолжение на сл. стр... }

```



Гн не вт

ШИФР НЕ ЗАПОЛНЯТЬ! ⇒

SG 81-92

Задача 5.

```

for i:=1 to m do {выводим промежуточных знач.
begin
  for j:=1 to Kn do {выводим умножаем на каждую цифру
  begin
    p:=0Ej r
    for l:=1 to k do {в этом числе мы умножаем b[i]
    begin
      P:=b[i].a[l] div 10;
      a[l]:=a[l]+a[i].b[l] mod 10+p;
      if (a[i]>=10) then {если a[i]>10
        p:=p+1;
      end; ps:=p; end;
      if (ps>0) then
        begin
          while (ps>0) do
            begin
              if k>r then r:=r+1
              a[r]:=ps mod 10;
              ps:=ps div 10;
            end;
          end;
        end;
      for i:=r down to 1 do
        write(a[i]);
    end.
  
```

{  
если  $r < 0$ , то нам  
нужно расширить на-  
ше число и умножать  
цифра знач.  $r$  делим на  
разряды и умножа-  
ем ~~и~~ их знач.  
в нам массив }  
  
{  
в результате мы получим от-  
вет но он будет перевёрнут,  
исходя массив a в сор. порядке }  
  
Задача 6.

Информацию о трансляции рассадки мы будем хранить в списках. У нас их будет ~~один~~ хранить (внешний) список хранит информацию о списках в  $L$  км. Второй (вложенный) будет хранить информацию о списках в  $K$  км. Мы будем проходить по этим спискам и если найдём это, то будем анализировать количество ярусов. Затем мы также массив с информацией от  $M$  до  $N$ . И, проанализировав кол-во ярусов у временного списка, прибавим +1 к тому же элементу массива, индекс которого равен кол-ву ярусов у этого списка. В результате мы получим массив, содержащий информацию о списках с некоторым числом ярусов. После выполнения ~~и~~ информации проверим её на корректность.  $M, N, K$  и  $L$  должны быть больше 0.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

102**СВ 59-65**

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

**ФАМИЛИЯ** Космин

**ИМЯ** Михаил

**ОТЧЕСТВО** Юревич

**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.



N4

program Z1;

var N, K, S, d, K2, S1: longint;

begin

readln(N);

K:=0; S:=0; d:=0; K2:=-d;

if N=1 then writeln(4) else begin

while S<N do begin

K:=K+1;

while (S<N) and (d<2) do begin

S:=S+K; d:=d+1; S1:=S1+d; K2:=K2+K; end;

writeln(S1\*3+K2+4);

end; end;

readln;

end.

(+)

N6

Считаю, что ини-бо парты, кеморое будем сохранять, заранее было известно и все данные по партам были записаны в таблицу бега:

Nомер	результат 1	результат 2

Записаны так, что все данные об всех сохранившихся партах были ровно переданы в таблицу; все строки абсолютно одинаковы по размеру.

Номер, начиная с которого все ини-бо очков в каждой парте ~~стали~~ стоят результатов 1, а затем стоят результатов 2, составил отклонение между найденными суммами; четвертое число будет соответствовать иронии, выраженной ини-бо очков, то и не может быть.

(+)



N5

Присвоить некоторой переменной I значение результата, стоящего на первой строке таблицы, сравнить его с введенным пользователем, и, если сравниваемый результат <sup>равняется</sup> значению переменной, то присвоить переменной значение результата; Переменной II присвоить значение суммы. Так, дойдя до конца таблицы, пересчитать строку, в которой стоит результат, присвоенный переменной, на первом месте, и вывести эту строку вниз. Присвоить переменной II значение переменной II ~~иначе единица~~.

При нахождении присваиваемой переменной I, первоначально результат брать из строки с номером  $n$ , где  $n$ -значение переменной II. Номер строки- её номер, при нулевом значении строки выразить при достижении переменной  $n$  значения номера на единицу, при ненулевом значении

N3

program 23;

var a,n,b,i,k: longint; label 1;

begin

readln(a,n); b:=-1;

k:=1;

for i:=1 to n do ~~begin~~

if ((a mod i)=0)and((n mod i)=0) then k:=k+1;

if k&gt;0 then begin writeln("Не 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 writeln("Не существует") else writeln(b);~~

1:

readln;

end.

(—)

(+) (7)



N2

Применение двоично-десятичной системы счисления бинарно-символично.

- 1) Требуется узнать каково число в записи числа десятичного вида, ее перевести во сколько в двоичную систему счисления.
- 2) Если требуется выделить какой-либо разряд числа, то это можно сделать ее переводе всего числа в двоичную систему счисления, перевести чисто четырехбитный двоичный код нужного разряда.
- 3) Более легкий перевод из бинарной системы как и из двоично-десятичной т.е. в десятичную, так и обратно, возможен в двоичную в таблице соответствия строк.

N3

program 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:=l to K do
    D:=D*8
  S:=S+D
end;
```

—

=> Перевести ~~восьмеричное~~ число из восьмеричной с.с. в десятичную.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 713

*SГ81-97*

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ КУПРИЯНОВ

ИМЯ Кирилл

ОТЧЕСТВО ИГОРЕВИЧ

Дата  
рождения 24.05.1998

Класс: 11

Предмет: ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

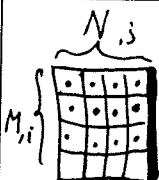
Работа выполнена на 2 листах

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:**  
ищем макс.  $\tau_{1-t}$ ,  
богатство цикла  
и пока не дойдёт  
до след. шага цикла.



1) Все числа с блокнотного места заносят в матрицу  $X (M \times N)$ , где левой верхний  $\tau_{1-t}$ -й блокнота будет в строке  $1, 1$ , матрица.

2) Найти максимальное  $\tau_{1-t} R$  и минимальное  $\tau_{1-t} Q$ . Поменять местами  $\tau_{1-t} R$  и  $\tau_{1-t} Q$ , если номер  $(i, 1)$ . Поменять  $Q$  и  $\tau_{1-t} (M; N)$ .

3) Начинаю со второго столбца, 1 строкки начинаем проверять элемент  $X (i, j)$  равен  $R - (k+1)$ . Если такое число найдено, то его нужно поменять местами с числом  $Q$ , обнуляем после  $R$ . После нахождения обнуляем место, поменять его местами, стоящими на 200-м месте  $R$ . Модифицируем  $Q$  и  $\tau_{1-t} (M; N)$ .

4) Проверяю до тех пор, пока убавление не дойдёт до числа  $Q$ . Тогда упражнение.

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

Пусть поле  $K \times L$  предстоит решить собой матрицу  $(K \times L)$  с некоторыми числами, значение которых = 0, т.е. по условию, не все они есть единичные. Найдём расположение максимума  $A$ .  $A [1] := B (1, 1)$ , массив  $A$  должен быть равен по количеству строкам числами, временно заполнив матрицу  $B$ . Если попадаете значение 0, то его числами в массив  $A$  это нужно заменить, когда массив  $A$  полностью заполнен, берём  $A [1]$  и снимаем с него ячейк в матрице  $B$  равной максимуму  $A [1]$ . Сколько настолько, сколько есть в матрице  $B$  ячеек. Тогда в  $A [2]$  содержатся не виды есть, а кол-во видов есть виды  $M$ . Аналогично, продолжаем эту же процедуру с  $A [2], A [3], \dots, A [N-M+1]$ . Затем возводим к ней правило: "Если вид  $M$  есть  $A [1]$ ,  $A [2], \dots, A [N-M+1]$ ". "Если вид  $(M+1)$  есть  $A [2]$ "

• Если вид  $N$  есть  $A [N-M+1]$ . Постановка.

№ 1

1) Продолжим, знаяется ли свободное место Реболати. Если да, то возводим 1 и симметрично цикл, сколько чисел до этого доказано. Всего если нет, то нужно выложить из свободного места  $N$  по единице до тех пор, пока не дойдёт до блокнотного места  $Q$ .

2) Всеми из  $N$  местами и продолжим где полученных числа  $\Phi$  (если из которых полученных чисел должно быть  $n$ , то она не дойдёт до бл. числа  $\Phi$ ).

3) И так далее пока не погребется. Т.е. числа  $\Phi$ , которое из всех мест, содержит занятое. В массив, например,

ПРИМЕР: ввели 25.

$$\begin{array}{r}
 24 \\
 23 \\
 22 \\
 21 \\
 \hline
 25 - 21 = 4 \\
 \hline
 4 - 3 = 1 \\
 1 - 1 = 0 \\
 \hline
 25 = 1 + 3 + 21
 \end{array}$$

4) На первых свободных числа заносят симметрично таким, где есть уменьшение чисел среди  $T-L$ . 1 и 3 и 21. 1 стоит на 1 месте, 3 стоит на 3 месте, 21 стоит на 7 месте. Соответственно, при 1, 3 и 7 местах ставят единицу в начальном числе, а на 2, 4, 5, 6 - нуль.





SF 81-97

(14)

Очевидно, результатом будет представление из сей формулы числа  
Такое число умноже хранить в памяти. Число представить в виде  
целого числа (наименьшее) в степени ~~базисной~~<sup>базисной</sup> ... , что  
умножение будет спрощено. Например,  $3^8 = 3^2 \cdot 3^2 \cdot 3^2 = 3^{2+2+2} = 3^3 \cdot 3^3 = 3^6 \cdot 3^2 = 3^8$ .  
Следовательно, наименьшее нормальное, записанное результатом в памяти  
A. первую цифру в первую ячейку, вторую - во вторую и т. д.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7092**ZE 51-12**

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:

Лысков

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

$$1 \cdot 1! + 2 \cdot 2! + \dots + n \cdot n! + 1 = (n+1)! \quad \text{при выполнении условия}$$

*тогда что можно было только вырасти через эту систему с. и только 1 способом*

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

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

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

c. с. является позиционной.

⊕

Ответ: c. с. f является позиционной.

№2

T - ГМТ

P - поле

K - кодирование при переводе чисел, может быть повторяющимся, ограниченным в 0.

T

P

⊖

C	C	C	C	C	C	C
K	K	K	K	K	K	K

$$t_{pc} = T + P + K$$

№3

```

var p, i, a, b, k : longint;
begin
readln(p); 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.

```

±

Многоязык - Pascal



№5

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

$$100000000 : 77 = 129 \cdot 870,129 \Rightarrow \max = 129 \cdot 870,129 = \\ = 89999977$$

var i;

begin

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

while i &lt; 100000000 do begin

i := i + 77; ↙

writeln(i); ↙

end;

end.

(5)

№6

$$1 + \epsilon_M = 1 \quad \text{зкл} \quad \text{помимо} \quad \text{член} < \epsilon_M, = 0.$$

var e, c: real;

begin

e := 1; c := 1;

while e &gt; 0 do begin

c := e;

e := e/2;

end;

writeln(c);

end.

(6)

№7

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

begin a[1]:=1; d:=1;

for c := 1 to 44 do begin

for k := 1 to d do begin

{ a[5k+1]:=a[5k+1]

a[5k+1]:=a[5k+1]+a[5k] div 10;

a[5k]:=a[5k] mod 10;

if a[5k+1]&gt;0 then d:=d+1; end;



End;  
~~while~~ ~~a[k] > 0 do begin~~  
~~k := k + 1;~~  
~~K := d down to 1 do begin~~  
~~if a[~~k~~] = 0 then K := d;~~  
~~while a[~~k~~] = 0 do begin~~  
~~d := d - 1; K := d; p := 0~~  
~~for d := K down to 1 do begin~~  
~~if a[d] = 4 then~~  
~~p := p + 1;~~  
~~end;~~  
~~writeln(p);~~  
 End.

~~(F)~~

N/4

$$N = \overbrace{a_1 b_1 c_1 \dots x_1 y_1 z_1} \quad k := 1$$

$$M = \overbrace{a_2 b_2 c_2 \dots x_2 y_2 z_2}$$

$$N = a_1 \cdot 10^x + b_1 \cdot 10^{x-1} \dots + y_1 \cdot 10^0 + z_1$$

$$M = a_2 \cdot 10^x + b_2 \cdot 10^{x-1} + \dots + y_2 \cdot 10^0 + z_2$$

~~$$z_1 \cdot z_1 + z_1 \cdot y_1 \cdot 10 \dots + a_1 \cdot a_1 \cdot 10^x$$~~

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

$$N[1..1000] \quad C[x] = CX$$

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

$$C[x+y] = K[x] \cdot d[y];$$

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

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

$$K := 1$$

~~for i := 1 to M do~~

$$K := K \cdot N \quad \left\{ \begin{array}{l} \text{отделить} \\ \text{остаток} \end{array} \right\}$$

~~writeln(K);~~

End.

$$\left. \begin{array}{l} C[1000..1] \\ M \text{ раз} \end{array} \right\}$$
~~(T)~~

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

BC 25-17

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ МАКСУДОВ

ИМЯ АНДРЕЙ

ОТЧЕСТВО ВЛАДИМИРОВИЧ

Дата  
рождения 30.10.1996

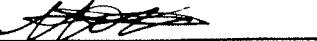
Класс: 11

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 обходить 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;~~  
~~{можно было обходить без вспомогательной переменной}~~  
~~{c: a1 := a2 - a1; a2 := a2 - a1; }~~  
~~until a1 = 0~~  
~~end; end.~~  
~~{Язык программирования: Pascal!}~~

2. ~~var~~  
~~j, i, p, s, d, c, a, t: Int64; e: boolean;~~  
~~begin~~  
~~read(p); if p ≤ 2 then write('простка 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 > c~~ do begin~~  
~~c := c \* 2; inc(t); end; dec(t);~~  
~~{считало, что t ≤ log<sub>2</sub> p}~~  
~~{t можно было присвоить 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 случаев}~~  
~~{if c mod p = 0} {проверка составности}~~



```

else begin
  for r:=1 to s-1 do c := (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 begin e := true;
    else write('Число простое');
    writeln('Составное');
  end;
  if e then write('составное')
  else write('Число простое');
end;

```

5. После чтения матрицы (для которой ранее выделены память), проходили по строкам матрицы и заносили числа в массив ~~чисел из строк~~  
 (каждая строка - это число, сформированное из заданных чисел в матрице; в нулевой клетки матрицы занесено 0, а клетки с '-1' - число -1)  
 ('число -1 будет заноситься в строку как символ '-'  
 а число 10 будет считаться пустой клеткой при проходе по матрице). Инициализация следующая:

Var b: ~~array~~ [1..M, 1..N] of byte;  
 a: array [1..(M+1) div 2] \* N] of string;

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

```

procedure qsort (a: mas, first, last: longint);
Var l, F: 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 l > F;
    if l > first then qsort(a, first, l);
    if l < last then qsort(a, F, last);
  end;

```

Далее выводим массив  $a$  и, если возможно, выведем несколько чисел в одну строку (обычая форма



чисел  ~~$a$~~  +  ~~$b$~~  и  ~~$c$~~  не должны превышать  ~~$M$~~  (также не превышать  $M$  ~~не~~ должно)

6. На  $1 \text{ км}^2$  приходится  $5 \cdot 10^5$  саженцов.  
Хранить данные будем следующим образом:  
 $a: \text{array}[1..K, 1..L] \text{ of } \text{сог};$   
предполагается что  $\text{сог}$  имеет тип  $\text{сог}^n$ :

т.у.р.е

$\text{сог} = \text{record}$

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

~~FOR  $e := 1$  TO  $N$  DO~~

{long int  $b$  задан так как саженцов  
не более  $5 \cdot 10^5$ }

~~FOR  $i := 1$  TO  $K$  DO~~

~~FOR  $j := 1$  TO  $L$  DO FOR  $e := M$  TO  $N$  DO~~

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

(В массиве  $b$  хранится количество саженцев каждого вида  
на ~~данной конкретной~~  $\text{стеке } 81 \text{ км}^2$ )

( $s$  - массив, сумм, т.е. общего количества всех  
каждого вида, который после окончания цикла  
нужно вывести)

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

4. Если ~~числа  $b$  не стали большими~~, то достаточно  
одного обр.

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

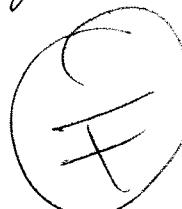
~~if  $N=1$  then write(1) else~~

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

~~а дальше цикл от 1 до  $M$  для  $C := N$~~

~~FOR  $i := 1$  TO  $M$  DO  $C := N$~~

(длинная арифметика - умножение)



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7102**AN 49-64**

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ МихайловИМЯ НикитаОТЧЕСТВО МаратовичДата рождения 27.08.1992Класс: 10Предмет ИнформатикаЭтап: ЗаключительныйРабота выполнена на 3 листахДата 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.



## Олимпиада школьников «Надежда энергетики»

ВАРИАНТ 7102

ШИФР НЕ ЗАПОЛНЯТЬ! ↳

AN 79-64

N<sup>o</sup>4

```

Var a,n,i,s: integer;
begin
  readln(n); a:=1; s:=0;
  while 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.

```

N<sup>o</sup>5

```

Var n,kod,k,a,b,s: integer; a,s,c:string;
Mass: array[1..100] of Record
  param: string;
  rez: integer;
begin
  readln(n); a:=' ';
  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; imax := i;
    for k:=i to n do if Mass[k].rez > max then max := Mass[k].rez;
    begin
      max := Mass[k].rez;
      imax := k;
    end;
  end;

```

N<sup>o</sup>3 - test



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

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

N<sup>o</sup>1

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

```
function powerstep(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('D');
```

~~b:=b mod 500;~~

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

~~for i:=1 to 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 i:=1 to b mod 5;~~~~if b=1 then write('I'); if b=2 then~~~~for i:=1 to b mod 1 do write('E');~~

```
end.
```

(F)

N<sup>o</sup>6

Var Mass1, Mass2 : array [1..100] of integer  
i, S1, S2, n : integer.

begin  
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.

(7)

N<sup>o</sup>2

Когда нам нужны доступ к членам, и, чтобы не переводить  
из 2cc в 10cc ~~и~~ обратно, используем глобальные переменные. т.е.

(1)

Выдан дополнительный лист: 1

Олимпиада школьников «Надежда энергетики»



№ группы

Вариант №

7102

*АН 79-38*

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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 Mass: array [1..4] 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 downto 1 do
if Mass[i]<> '' then Write(Mass[i]);
end.

```

②

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



③

```

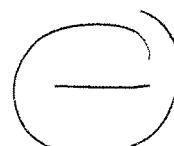
Var a, b, c: integer;
Begin

```

```

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 однозначны')
else Writeln(c);
end.

```

④ Var A: array [1.. 10000] of record  
 zh: string;  
 rez: integer;  
 end;  
 i, max, k, N: integer;



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

(+) (7)



(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('Danui');  
end.  
if S > S1 then Write('Temp')  
else if S = S1 then Write('норма')  
else Write('Danui');  
end.

7

## Олимпиада школьников «Надежда энергетики»

<input type="text"/>	<input type="text" value="WU 36 -34"/>	← Не заполнять Заполняется ответственным работником
№ группы	Вариант №	шифр
	<u>7092</u>	
ФАМИЛИЯ	<u>Николаев</u>	
ИМЯ	<u>Семен</u>	
ОТЧЕСТВО	<u>Александрович</u>	
Дата рождения	<u>08.09.1999</u>	Класс:
Предмет	<u>информатика</u>	Этап:
Работа выполнена на	<u>3</u> листах	Дата выполнения работы:
<u>27.02.15</u> (число, месяц, год)		

Подпись участника олимпиады:

НС

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



7. program z7;

```

var a:array [1..100000] of integer; t,n,i,s,j:longint;
begin
  a[1]:=t;
  t:=1;
  n:=0;
  for i:=2 to 44 do
    begin
      s:=t;
      for j:=1 to s do
        begin
          a[j]:=a[j]+n;
          m:=a[j]*i;
          a[j]:=m mod 10,
          h:=m div 10;
          if a[j+1]=0 then a[j+1]:=n;
          if a[j+1]>=0 then t:=t+1;
        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, ' ');
      writeln;
    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, ' ');
  writeln;
end.
```

~~всегда~~ всегда остаток от деления на 4 в массиве.

~~7~~

определение номера разрядов, содержащих цифру 4.  
" " - означает пробел

5. program z5;

```

var t:longint;
begin
  t:=10000 div 44;
  t:=(t+1)*44;
  writeln(t);
  while (t<=999999999) do
    begin
      t:=t+44;
      writeln(t);
    end;
  writeln;
end.
```

~~5~~

находит первое число, которое делится на 44, завершает программа. Заменить можно на 1000000000.

Данное бодягум число из промежутка  $10000 \leq x \leq 999999999$ , при этом число, которое бодягум, кратно 4 и 11.



```

3. program Z3;
var n,i,j,s,t: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+j;
        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  $N$ , чьи суммы делются на эти числа.

Внешне программа и не грамматична, т.к. она не содержит явного числа.



2. Следует разбить все страны на два кампания с неравным временем и без перевода времени.

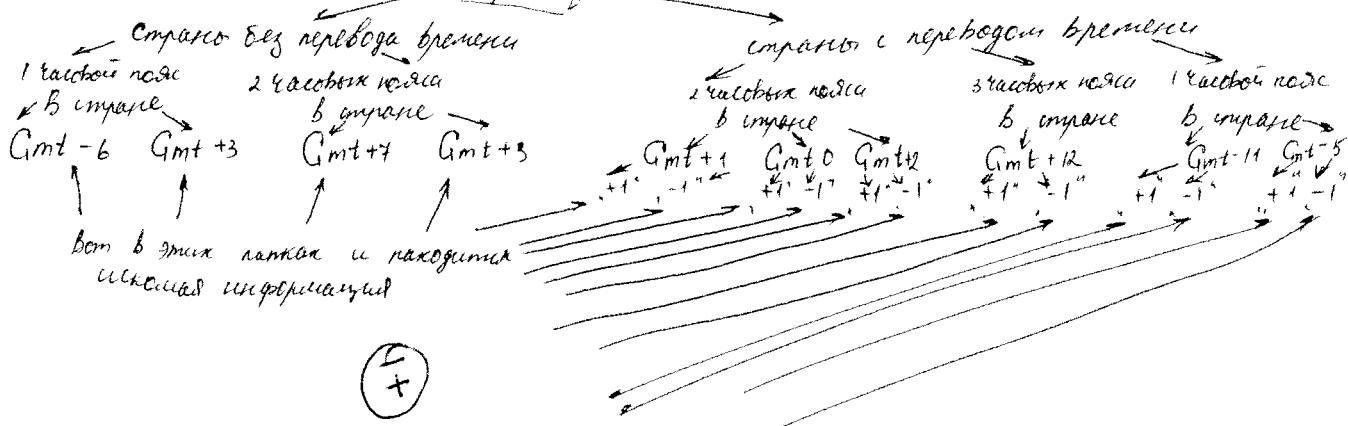
Рассмотрим группу стран без перевода времени. Этот кампания разбита на количество часов в стране. Далее разбивается кампания на количество часов в стране на часовые пояса, в которых будет устанавливаться время в каждой стране.

Время в группе стран с неравным временем. Аналогично разбиваем этот кампания на группы по количеству часов в стране. Сейчас представим часовое пояса этих стран если бы время не было во переводе. Разбиваем кампанию по количеству часов в группе на часовые пояса, которые могут сейчас изменяться. В кампаниях из полученных кампаний создадим еще по две строки изменения:  $+1''$  и  $-1''$  (это означает перевод времени). В этих группах и будем хранить информацию о местном времени в указанной стране.

Представим все это в дерархической системе (для примера)

разбив исходной кампания время

время //





4. 1) Запишем все числа в массив. { например число 1632841.

2) начнем с конца массива брать из второй строки ~~один~~ цифру, умножая разрядность

в нашем примере число 1632841 можно умножить на 1 (разрядность), затем на 4, 8, 2, 3, 6, 1. (не забудьте умножение!) делаем эту операцию, как при вычислении разности в загадке.

$$m := a[1, j] := a[1, j] + n;$$

$$n := a[1, j] * \alpha[2, i]$$

$$a[1, j] := m \bmod 10;$$

$$n := m \bmod 10;$$

$$\text{If } a[1, j+1] = 0 \text{ then } a[1, j+1] := n.$$

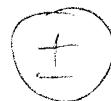
Во второй строке то же самое

Вот так проходит весь первый ряд.

Умножив по ~~один~~ через нули конец первого ряда на 0. Теперь еще раз, что начинаем проходить ряд сева направо.

Число первого ряда означает (в нашем случае) первую строку умножаем начиная с 1, потом на 4, 8, 2, 3, 6, 1 (умножение разрядность!) а [1, i] начинает перечислять выше засчитан.

Делаем это прошло 14 раз.



## Олимпиада школьников «Надежда энергетики»

№ группы      Вариант №      7113

SG81-20

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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.



Для 4  
Баллов  
20

Пусть  $K \leq L$ . Тогда если известны, то будем использовать массив размером  $\left[\frac{K}{2}; \frac{L}{2}\right]$  (мк К и L будут введенны, а дальше есть замыслы на более  $2 \times$  ~~как~~ масив). Далее создадим массив размером  $[N-M+1]$  (который будет отвечать за кол-во различных единиц на ноле). Далее будем заполнять массив нолями, сразу же добавляя не единицы, если встретим единицу в ячейку = ярусу единиц. Затем, начиная с  $i=0$  до  $(N-M-1)$ , будем накапливать: "ярус" номер ~~единица~~ и ее значение, если значение ячейки  $\neq 0$ .

Алгоритм:

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

Считаем значения  $K; L; M; N$

Создадим массив  $Mass\left[\frac{K}{2}; \frac{L}{2}\right]$  (округление деление

в большую сторону, ~~чтобы~~ целой части)

Создадим массив  $Znach[N-M+1]$ .

Заполним  $Mass$  циклом  $i = 0$  до  $\frac{K}{2}$

в цикле ~~ненулевые~~ ненулевые единицы  $j = 0$  до  $\frac{L}{2}$ .

если на ноле  $\neq$  единице  $[i;j]$  нуло, то

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

иначе

$Mass[i;j] = \text{ярус дерево}$

$Znach[\text{ярус дерево}] = Znach[\text{ярус дерево}] + 1$ .

конец если.

иначе цикл

иначе цикл

цикл  $i = 0$  до  $(N-M+1)$

если  $Znach[i] < > 0$  то

напечатай: "ярус:" +  $i$  + " Кол-во:" +  $Znach[i]$

иначе если ноль.

иначе ноль.

$N=3$ -стр



N5 Создадим массив размером  $[M; N]$ , затем считаем строки массива с помощью блокнота, затем для уменьшения кол-ва работы, воспользуемся кол-вом строк, символов которых  $\neq " "$  (не пустой), заменив каждую строку в переносную  $"Q"$ , затем заменим все строки разделённых  $" "$  (пробелом) в одну переноску.

типа строки: Создадим список и будем добавлять в конец значения каждого раз, когда будем отыскивать в строке новое число, заканчивающееся, когда после цифры, будет два крошащих пробела (пробела). Затем на-то  $Y[i]$  в списке в переносную  $Z$ . Пройдём циклом от 1 до  $Z-1$  и в нем циклом от 1 до  $Z-1$  отсортируем по убыванию (сравнивая текущий и следующий элементы списка, заменяя в массиве изменения списка).

Затем циклом от 1 до  $Z$  заменим все значения массива.

АЛГОРИТМ: Создадим чистоисчислительные переменные:  $Z; M; N; i; Y; K$ .

Создадим переменную типа стока :  $Q$

Считаем  $M; N$ .

Создадим массив  $Mass [M; N]$ .

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

Циклом  $i = 0$  до  $M$

Найдем кол-во строк, где первым символом  $<> " "$  конец строки.  
Заменим получившее значение в  $Y$ .

~~Затем  $Y$  записываем в строку с  $i = 0$  до  $Z$ , меняем  $Y$  на  $Y + 1$ .~~

Числом  $i = 0$  до  $M$  ~~и  $Y = 0$~~   $N$

Числом,  $j = 0$  до  $Y-1$

Если  $(j+k \text{ символ } + j+1+k \text{ символ}) < > " "$ , то

~~Конец внутреннего цикла.~~





Если  $j \text{ выходит за } "0" \text{ то } k=+1$   
 Заносим ~~значение~~ в конец списка  
 идем

$K = K + 1$  (К позволяет нам залатать ошибку списка

конец цикла

конец цикла

конец цикла

значение  $i = 0 \dots n - 1$  из массива  $Znach$  с индексом  $k$  заносим в список

конец цикла.

Чтобы создать массив  $Znach$  из  $i$  списка в массив

перекинем значение  $Znach[i] = i$  списка в массив

конец цикла  $K = 0$

чтобы  $i = 0 \dots Z - 1$

чтобы  $i = 0 \dots Z - 1$

Если  $Znach[i] < Znach[i + 1]$  то

$K = Znach[i + 1]$

$Znach[i + 1] = Znach[i]$

$Znach[i] = K$ .

конец цикла

конец цикла

конец цикла

Оформим все цикла по убыванию, используя цикл

Цикл  $i = 0 \dots Z$

Написать в строку  $Znach[Z]$

конец цикла.

№ 1 Считали исходное число в переменную  $"Q"$

Затем будем сравнивать  $Q$  с числом Рибонки

которое не найдено, которое больше  $Q$ . Занесение  
 номер этого числа-1) в переменную  $N$ , затем  
 создание массива (~~заполнения~~) с элементами  $[N]$ , после

этого будем вычитать из  $Q$  число Рибонки ног

номером  $i = 0 \dots N - 1$  (изначально посчитали и занесли  
 в массив Рибонки в массив  $Znach$ )



Если  $Q > \text{знач}[i]$ , то будем заполнять в ячейки массива 1 иначе 0. Таким образом заполнив массив 1 и 0, отразим его на экране.

Алгоритм:

Создадим начальные 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 цикла;  $Y=1; Y_1=1; Y_2=1; N=N+1$ ; создадим массив  $\text{Mass}[i]$

циклом  $i=0$  до  $N$

~~Mass[i] = Y~~

$$Mass[i] = Y$$

$$Y = Y_2 + Y_1$$

$$Y_1 = Y_2$$

$$Y_2 = Z$$

конец цикла  
создадим массив  $\text{знач}[N]$ .

циклом  $i=0$  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 цикла.

Конец цикла.

циклом  $i=0$  до  $N$

напечатаем в строку. заарене значение  $\text{знач}[i]$ .

Конец цикла.

Чтобы  $\text{III}.$  число  $N^m$  можно было вывести на экран, то можно





использовать списки, т.к. кол-во цифр ~~ибо~~ получившееся число не существует. Тогда можно представить каждую цифру, как новую ячейку списка и в другую ячейку, умножаем  $N$  на  $N$ , раз. Второй способ заключается в том, что можно вдвое уменьшить массив, типа long и умножать  $N$  на  $N$ , раз, до того момента, пока это не превзойдет значение (максимальное) типа long. Затем перейти в другую ячейку массива с соответствующим переносами чисел. Третий способ заключается в разделении  $N^m$  на  $\boxed{N^2}$  и дальнейшей ручной работой в текстовой переменной (слева вручную, ~~оператор~~  
здесь "если")

(+)

N2. Создадим два массива, размером  $10 \times 10$  всех комбинаций пару из четырех чисел в промежутке от 1 до 60. В первый массив занесем все числа (складем их) и переберем варианты одинаковых квадратов чисел, а во второй будем заносить в первый способом. Или, во второй кол-во раз, которое оно повторяется.

Затем будем искать числа, которые встречаются, при переборе массива из первого способа массива и сам значение второго столбца в той же строке, меньше переменной, которая складывает максимальное кол-во представлений, то занесем в неё значение ~~столбца~~ второго столбца, а в другую переменную значение первого столбца той же строки. Там самую же найдем необходимое число  $N$ .

(+)

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

4102**CW 59-50**

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ ЛУГРАЧЕВА  
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:

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.

тигаде 1 год. лист

Му

Длина чисел, записанных в римской системе счисления, велика: такое представление 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 (сохраняя 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 возможный (в Паскале int64, например) 3) символ 2)  $n$ : тип данных такой, что можно считать цифру (в Паскале byte)

$\Rightarrow$  фрагмент программы на языке Паскаль:

```
READ(c); if c='-' then READ(a) else
while not (eoln) do
```

```
begin
  READ(n);
  a:=a*8+n;
end; <<
```

Если  $c=-$ , то после ввода  $a$  нужно ввести  $-$ , 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"); <<
```

Объяснение: переменная  $n$  использовалась при считке, 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  $B[1..3, 1..3]$ , такой, что

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

Далее основная часть второй подзадачи:  $B[3,1]=I$

$\Rightarrow$  фрагмент программы на языке Паскаль

```
for n:=1 to 3 do
```

```
begin
```

```
case m[n] of
```

```
1: write(B[n,1]);
2: write(B[n,1], B[n,1]);
3: write(B[n,1], B[n,1], B[n,1]);
4: write(B[n,1], B[n,2]);
5: write(B[n,2]);
6: writeln(B[n,2], B[n,1]);
7: write(B[n,2], B[n,1], B[n,1]);
8: write(B[n,2], B[n,1], B[n,1], B[n,1]);
9: write(B[n,1], B[n,3]);
```

```
end;
```

```
end; <<
```

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

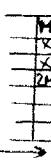
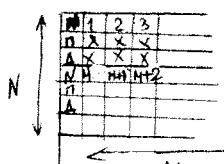
Если  $a$  - основное число, то { }, иначе фрагмент программы выше.



Учитывая, что "особых" чисел ограниченное количество, можно в начале программы создать массив соответствий: "особое число — его римская запись. Тогда на месте слов "если а — особое число..., то" будет проход по ячейкам массиву.

Задание 6.

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



1) Т.е. они начинали записывать результаты с первого верхнего угла, начиная с строки и горизонтально могли записать результаты  $(M-1)$  игр, затем они переходили на строки 4-6 и т. д. При этом вертикально можно использовать  $(N \text{ div } 3)^* 3$  строк, чтобы для каждой группы игр имелись остаточные  $N \text{ mod } 3$ .

(Если  $N$  при делении на 3 дает в остатке 2, то в конце может быть записано  $N \text{ mod } 3$  и  $M$ , а результатами Данила не хватит места). При этом  $M$  — любое число, большее 1, чтобы этот листок вовсе не содержало информации.

Чтобы этот листок содержало 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  $S$  сохраним количество трехстрочных блоков ( $S := N \text{ div } 3$ ). 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; <<
```

Используемые переменные:

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

В результате работы этого фрагмента программы в переменной  $sp$  будет серия трехзначных оценок, в переменной  $sd$  — сумма баллов Данила остается 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')
```

Задание 5

Количество строк введенного файла не известно  $\Rightarrow$  делаем программу "пока не конец файла".

Считываем построчно (в строке изначально номер, значки, результаты). Удаляем до первого пробела — номер, он нам не нужен, его никак не сохраняется. Удаляем все до позиции пробела включительно. Далее до следующего пробела (не включительно) имеет значение параметра, выделяем эту часть так же с помощью функции поиска позиции, сохраняем в массив патр[i] из строк. Удаляем из остатка всех до позиции пробела включительно. Там остается только результат, но он представлен как строка, а не как число. Переводим его в число с помощью функции val (на языке Паскаль) или т. п. и сохраняем в массив res[i] из чисел (следует учсть, что результаты могут быть большими, поэтому нужно выбирать большой тип данных). Так же при вводе считаем количество введенных строк ( $N$ ). Далее расположим результаты в порядке убывания (от большого к маленькому), 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 цело  
типа;

$s$ -переменная типа  
string.



В конце от 1 до N нужно выводить

1) номер строки

2) name [номер строки]

3) res [номер строки],

вывод будет происходить в порядке убывания, что и требовалось в задании.

⇒ фрагмент программы на языке Паскаль:

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

```
  write(i, " ", name[i], " ", res[i]);
```

Номер  $i$  будет соответствовать строке, результатом которой  
является на  $i$ -мой месте по убыванию, а не том, который  
приводится  $i$ -мой по строке. Если просто сохранить номер строки  
то получается трёхзначный массив  $n[i]$ , при сортировке в куче  
будет сохраняться всё до первого пробела, при сортировке с кучей  
будет происходить только то же самое, что и с массивом  $n[i]$ .

#### Задание 4

Если пытаться делать то же, что делает робот, при этом каждый раз замыкая не обходящее количество рек для нового квадрата, то можно заметить закономерность:

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

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.

Возьмём переменную  $j$ -номер квадрата (или количество квадратов).

⇒ фрагмент программы на языке Паскаль:

```
i:=0; while i<1500 do
  begin
    j:=i+1;
    a[i]:=3;
    i:=i+j;
    a[i]:=3;
    incl(j);
  end;
```

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

Проходим по массиву еще раз (от 1 до 1500), если  $a[i]$  равно 0, т.е.  
в первый раз это не изменилось, то меняем значение  $a[i]$  на 2.  
Это всё происходит в программе каждый раз еще до считывания  
 $N$  (операции не так много, пользователь не заметит задержки). После  
считывания  $N$  считается сумма первых  $N$  элементов и она выво-  
дится на экран. Это и будет искомым количеством рек.

Задача 3*Был*

После счётывания  $n$  и  $a$  узнаем, являются ли они взаимно простыми:

- 1) Найдем меньшее из них операцией сравнения
- 2) от 2 до меньшего в цикле пытаемся разделить  $n$  на  $i$  — переменную — счетчик цикла. Если они оба делятся, то увеличиваем делаем разное true — так меньше защищается памяти)
- 3) после цикла смотрим на значение  $s$  (пусть она типа boolean, в начале программы была false).

Если её значение false, то не нашлось такого  $i$ , на которое бы делели  $n$  и  $a$ , и  $n \Rightarrow \text{НОД}(a, n)=1$ .

Если её значение true, то существует  $i$ ,  $i \in [2; \min(a, n)]$ , такое, что  $a; i$  и  $n; i$  одновременно  $\Rightarrow \text{НОД}(a, n) \neq 1$ .

4) 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 работы всей программы — сообщение „НЕ СУЩЕСТВУЕТ“! Иначе происходит выполнение п. 5.

5)  $\text{НОД}(a, n)=1 \Rightarrow$  мультипликативно простое число существует.

$\Rightarrow$  фрагмент программы на языке Паскаль:

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

Когда цикл завершится, переменная  $i$  будет иметь значение, которое мы можем найти ( $i$ -мультипликативно обратное число для числа  $a$ )

останется только вывести  $i$ .

(Структура программы будет следующей:

(описание переменных)  
начало  
если  $a$  и  $n$  — взаимно простые, то цикл поиску  $i$   
иначе вывод „не существует“  
конец.)

$i$  — это одна  
переменная  
типа boolean;  
 $i$  — целочисленная  
переменная, которая  
является счетчиком

Задание 2

Использование двоично-десятичной системы счисления уместно при

- 1) шифровании (простой способ, недорогой, но лучше, чем ничего)
- 2) 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)
- 3) если числа, с которыми работает программа, очень большие (больше, чем  $\approx 2^{10}$ ), то запись в двоичной форме будет занимать ~~非常多的~~ много места, или в двоично-десятичной (большее количество символов), ~~при увеличении числа в 2 раза в двоичной обозначимости~~

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

ИР 62-89

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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.



## Олимпиада школьников «Надежда энергетики»

Вариант: 7113

ШИФР, НЕ ЗАПОЛНЯТЬ!

ЧР 62-89

N1 a - исходное число в десятичной системе  
 $k, i$  - элементы  $d_{ij}$ -массива в единицах  
 б - система счисления в виде дательного  
 Начало ~~программы~~ алгоритма:

Входное число a;  
 $i=1;$

Любое  $b \neq 0$   
 начальное число

$$b := a : 10;$$

Берём целую часть от b;  
 $k = k + 1;$

конец цикла;

$$\begin{cases} d_{1j} = r; \\ d_{2j} = d; \end{cases}$$

~~Начало цикла~~

Дел j от 3 до i

начальное число

Любое  $a_{ij} < i$

начальное число

$$a_{ij} := a_{ij-1} + a_{ij-2}$$

конец цикла;

конец цикла;

Дел j:=0 и до i (отсчит от большего к меньшему)

Начальное число

$$\text{Если } b - a_{ij} > 0, \text{ то } b := b - a_{ij},$$

в 0 приписываем

справа  $1^{\text{у}}$

иначе в 0 приписываем справа  $0^{\text{у}}$

0 - отдаёт в рабоче.

Конец алгоритма.

N2  $b_{ij}$ -массив с вариантами преобразований  
 $a_{ij}$ -массив с квадратичные чисел  
 начальное алгоритма

Вход N1;  
 $k=0;$

Дел j:= от N1 до N1+50  
 начальное число

$$b := \sqrt{j};$$

квадратичные в

(-)

см. др. спр.

ANSWER

лист 01 из 04



Для  $i = от 1 по 8$   
 $a[k][j] = c^{i-2}$

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

$$B[k][j] = j^*$$

Если  $kmin > k$ , то  $kmin := k$

Если  $B[k][j] < 1$ , то  $B[k][j] := 0$

конец цикла;

Конец алгоритма.

Таким образом, в массиве началье массива будут находиться числа, которые можно представить в виде суммы четырех квадратов натуральных чисел начиная ссися конечное количество представлений.

№3 р-некорнейшее число

$$p1 := p - 1$$

Начало алгоритма

$$t := 1;$$

$$S := 0;$$

Вводим число p;

$$p1 := p - 1;$$

$$B := p1;$$

Пока остаток от  $p1 : 2^S$  не равен нулю

начало цикла

$$S := S + 1$$

конец цикла

$$d := \underbrace{p1}_{t} : 2^S;$$

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

округляем t

Для  $i = от 1 по t$ .

начало цикла

введенное случайное  $a$  из диапазона  $[1; p1]$

если  $a^d = 1$ , то введенное  $a$  вероятно простое?

См. АР. СФР.  $\rightarrow$





иначе выбираем случайное  $k$  из диапазона  $(0; s-1)$

$$k = d^r$$

если  $a^{k+d} = -1$ , то выбираем „вероятное”  
„правильное”

иначе выбираем „составное”;

конец цикла;

конец алгоритма.

№ 4 отв - ответ

начало алгоритма

Вводим  $M$ ;

Вводим  $N$ ;

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

для  $i := 0 \dots 1 \dots N$  и

$$\text{отв.} = \text{отв.} \cdot N$$

конец алгоритма.



№ 5  $a[i,j]$  - двухмерный массив с элементами, которые  
записаны в обратной  
по условию записи  $a[i,j]$  рано.

Начало алгоритма

для  $i := 0 \dots 1 \dots N-1$

для  $j := 0 \dots 1 \dots M-1$

дела  $a[i,j] < a[i,j+1]$

начало цикла

для  $i := 0 \dots 1 \dots N-1$

для  $j := 0 \dots 1 \dots M-1$

начало цикла

если  $a[i,j] < a[i,j+1]$ , то  $\delta := a[i,j]$

$a[i,j] := a[i,j+1]$

$a[i,j+1] := \delta$

конец цикла;

конец алгоритма.

СМ. АР. СГР. →



№ 6 Массив  $a[i]$ , где  $i \in [m, n]$

массив  $b[g, j]$  - результирующий, где  $g \in [1, \frac{k}{2}]$

$vet$  - количество ярусов верхок  $j \in [1, \frac{l}{2}]$

Нижний левый элемент массива  $b[g, j]$  - квадрат с предположением единиц

Начало алгоритма

для  $g := 0$  до  $\frac{k}{2}$

для  $j := 0$  до  $\frac{l}{2}$

если на  $a[g, j]$  есть единица, то  $a[g, j] = 1$   
иначе  $a[g, j] = 0$

для  $g := 0$  до  $\frac{k}{2}$

для  $j := 0$  до  $\frac{l}{2}$

если  $b[g, j] \neq 0$

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

Таким образом, данное о количестве ярусов  
каждого яруса будет вычислена так:

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

количество ярусов верхок

количество единиц яруса верх.

Данное о расположении единиц вычислится так:

k	0	1	1	0
	1	0	0	
	1	1	0	

если единица  
нет единиц

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7/13

SG81-27

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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:



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





Несколько наблюдений:

1) анл если  $N \leq \log_{10} A$   $\{1\}$ ;  $\{0\}$ ,  $\{1\}$ ,  $\{0\}$ ,  $\{N\}$ ,  $\{R\}$ )  
исх  $N$   
пер  $A$

нар  $y = N_1, N_2, R, y, y_m$

$R = \log_{10}(A)$ ;

$N_1 = N_2$

$N_2 = 0$

если  $N = N_1$  то  $R = 1$ ;

если  $N = N_2$  то  $R = 0$ ;

если  $R > N_2$  то  $R = R - N_2$ ;

иначе  $R = R + N_2$ ;

$R := N$ ;

иначе  $R > 0$ ;

$N_2 := 1$ ;  $N_2 := 2$ ;  $y := 1$

иначе  $N_1 \leq k$  и  $N_2 \leq R$

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

иначе  $N_1 := N_1 + N_2$

$y := y + 2$ ;

иначе если  $R = N$  то  $y_m = y$ ;

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

иначе  $A[\gamma] := false$ ;  $R := R - N_1$ ;

кнс

иначе если  $\gamma = 0$  то  $y_m = y$  иначе  
если  $A[\gamma] = true$  то  $R = R + 1$   
иначе  $R = R - 1$ )

кнс

если  $k = 0$

5) анл  $N \leq \log_{10} A$   $\{1\}$   $\{0\}$

исх  $A$

пер  $A$

нар  $y, max, N, dub, y_{max}, t$

если  $y < 0$

иначе  $R = \log_{10}(A)$ ;

$R = R + 1$ ;  $max := -30000000$ ;

если  $y > 0$  то  $N = R$

иначе  $R = \log_{10}(A)$ ;

$R = R + 1$ ;

если  $A[\gamma] > max$  то  $max = A[\gamma]$ ;  $y_{max} = \gamma$ ;

иначе  $dub := A[\gamma]$ ;

$A[\gamma] := max$ ;  $A[y_{max}] := dub$ ;

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

$t := 2$

если  $t > 0$  то  $N = 1$

иначе  $max := -30000000$ ;

если  $t > 0$  то  $N = 0$

иначе  $A[\gamma] > max$  то  $max = A[\gamma]$  иначе

$dub := A[t]$ ;

$A[t] := max$ ;

$A[0] := dub$ ;

иначе  $A[0] := max$ ;

если  $y > 1$  то  $N = R$  иначе  $A[\gamma] := 1$ ;

кнс



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7113

SG 94-97

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:

РК

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.)
 $\begin{cases} I=V, \\ K=I; \\ K>I, \\ M=0, \\ I=I, \end{cases}$ 
} - переменные для расчета числа  
Фибоначчи.
Пока  $i \leq n$ :

НАЧАЛО

В конец вектора (он изначально пустой) занести  
значение  $i$ .
 Затем  $M=K$ , }  $M$  и  $K$  меняют свои значения  
 $K=i$ , } и из них формируется следу-  
 $i=M+K$ ; } ющее число Фибоначчи  $i$ 

КОНЕЦ

Пока  $n < > 0$ :

НАЧАЛО

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

НАЧАЛО

 $r=r*10+i$  ( $r$  - переменная для  
хранения числа  $n$   
в представлении  
с.с. Фибоначчи)
 


 $i=n$  - число из конца вектора;  
удаляем число из конца вектора.
 

КОНЕЦ

ИНАЧЕ

НАЧАЛО

 $r=r*10+0;$ 

удаляем число из конца вектора;

КОНЕЦ

КОНЕЦ

ВЫВЕСТИ  $r$ .



## Задача 5

Матрица можно представить в виде 2<sup>х</sup> мерного массива  $M \times N$  клеток.

Тогда нужно выполнить следующие действия:

1) упорядочить по увеличению сева направо каждую строку массива

2) упорядочить по увеличению сверху вниз каждую столбец массива

3) упорядочить по увеличению каждую из диагоналей массива сверху вниз справа налево  
Эти действия нужно выполнять до тех пор, пока производится хотя бы одна перестановка.

После этого получим 2-мерный массив, упорядоченный по увеличению сева-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

5)

143	28	18
16	15	12
11	7	5
4	3	2

- конец программы

Сортировку можно производить например пузырьковым методом.



## Задача 6

Разобьем весь участок эксперимента на квадраты 1 км × 1 км.

Подсчитаем количество единиц каждого вида в квадратах из квадратов и заметим что в двумерный массив исподузе для хранения собственности данных.

Затем проделавшие по массиву и на квадраты числу прибавляя к общему числу единиц данного типа, число единиц этого типа в текущей ячейке получим общее число единиц <sup>данного типа</sup> на участке. —  
Тогда будем выполнять параллельное  
этот алгоритм повторяя не только подсчетами  
число единиц ~~по~~ каждого типа в общем, но и  
получим статистику распределения типов единиц  
по участку эксперимента.

Пример:

Вид участка сейчас:

	1	2	3	4
1	(5)	(6)	(5)	(6)
2	(7)	(5)	(7)	(5)
3	(6)		(7)	(7)
4	(5)		(5)	
5		(6)		(7)

Массив с информацией

	1	2	3	4
1	(5)-1	(6)-0	(5)-1	(6)-0
2	(6)-0	(7)-1	(6)-0	(7)-0
3	(7)-0	(7)-0	(7)-0	(7)-1
4	(5)-0	(6)-0	(5)-0	(6)-0
5	(6)-0	(7)-0	(6)-0	(7)-0

6) количество друзей  
Понятно, что если можно не растянуть строку в  
двумерных квадратах

тогда можно, что, как представившего выше,  
записывать ее в том квадрате, где распределяется  
её часть, а если эти части равны то  
предномерия отдаются верхней и левой квадра-  
там.

Второй вариант - это записывать в квадрат  
строки зеркально, соответствующее произвольному  
размеру единичного типа, находящегося в этой  
квадрате. Такой вариант позволяет получить до-  
лее точную статистику, но общее количество  
единиц от этого не изменится.



## Задача 3

Так как числа  $a$  вводятся случайно, можно предположить, что случайно введено  $t$  число  $a \bmod 0$  или  $0$ , чтобы не отменять ошибочные числа.

$t = \log_2(p)$  с округлением в большую сторону, чтобы не ошибиться.

Все простые числа, кроме 2 и -2 нечетные, так что если введенное число  $p$  четное и не равно  $2^{n-2}$ , то запускаем тест Миллера-Рабина тем числом и можно просто вывести "Составное" и завершить программу.

Если же число  $p$  нечетное (проверяется функцией  $\text{mod } 2$  на равенство 0), то  $p-1$  обязательно будет четным, а все четные числа кратны 2-и.

Тогда можно  $s$  присвоить 1, а  $d$  присвоить  $(p-1)/2$ . В случае если  $p$  нечетно, проверим равенство  $2^{d-2} \bmod p$  равно либо 1, а если нет, то программа

далее нужно вычислить  $t$  как количество единиц в числе от  $a=1$  до  $t$  проверять два условия.

Первое условие можно проверить так:

В числе от 1 до  $t$  в некоторую переменную  $m$  занести  $m^* a$ . Таким образом в  $m$  находится  $a^d$ .

+

Далее нужно сравнивать полученный результат ( $m$  из переменной  $m$ ) с результатом варанания  $1 \bmod p$ . Если они равны, то вводим. В противном случае, выходим из цикла и завершаем программу.

В случае если  $d$  было отрицательным то в переменную  $m$  занести  $m^* a^{\frac{1}{t}}$  и продолжать все тоже самое.

Второе условие можно проверить так:

Так как  $s$  уже равно 1, то  $r$  может быть только 0. Тогда достаточно проверить равенство  $a^d = 1 \bmod p$  по аналогии с первым условием. Если равенство выполнится, то вводим. В противном случае, выходим из цикла и завершаем программу.

Если ни одно условие не выполнится, то разделяем это можно проверить с помощью переменной  $sol$  и то нужно ввести  $"Составное"$  в маску завершения программы. Кстати, то алгоритм не повторяет в то время теста Миллера-Рабина, однако он не противоречит ему.



## Задача 3 продолжение

Если же одно из условий не выполнено  
ни разу (это можно проверить с помощью пе-  
ренесения - флага), то нужно вывести соответствую-  
щее сообщение.

Конечно, этот алгоритм не подходит в мож-  
ности места Михаила Рабина, однако он не  
противоречит ему.

## Задача 4

Для решения этой задачи нужно исполь-  
зовать длинную арифметику (каждая цифра  
числа  $M$  и  $N$  помещается в одну из ячеек ма-  
шины). Далее можно попытаться достичь заско-  
чительной скорости вычисления произ-  
ведения  $M \times N$  так как максимальные величины  
 $M$  и  $N$  это  $1000^{1000} = 1000 \cdot (10^3)^{1000} = 10^{3000}$ . Вероятно  
стандартные программы не создают массив такой  
размера, так что можно подготовить разбить его на  
несколько массивов и обработавшего их вместе.

Помимо следующего, используя длинную арифметику учи-  
тывая  $N$  на  $N$  до тех пор, пока массив с чистотой, каш-  
ает раз уменьшением на единицу, не однозначна.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ

Рубик

ИМЯ

Стамишав

ОТЧЕСТВО

ЕвгеньевичДата  
рождения14.04.1998Класс: 11

Предмет

ИнформатикаЭтап: Заключительный

Работа выполнена на

5 листах

Дата выполнения работы:

27.02.15

(число, месяц, год)

Подпись участника олимпиады:

Сергей

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

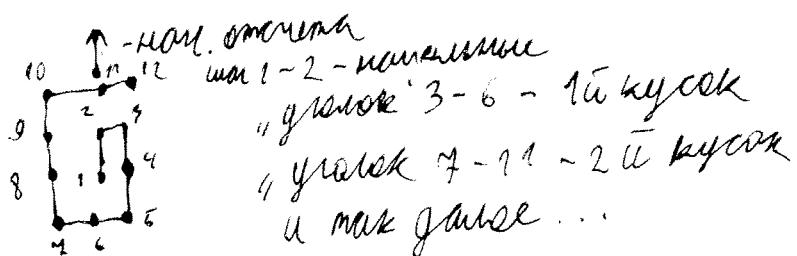
Было бы здорово, если бы  
было бы здорово, но...

Были бы в движении работы автоматизированной рабочей ячейки.

Возможна ли такая схема второй пакетической работы изображена.

Назовём куском - отрезок движения, блокирующий два робота.

(именно - тракторная работа)



Задача: что будет получено в начале цикла и на  
шаге  $k+1$  ( $N+1$ ) // 2, где  $n$  - длина куска. Всего от  $n=1$  до  $n=2^{k+1}$   
кусок имеет длину  $\frac{32}{2} \cdot \frac{3}{2} \cdot \frac{2}{2} \cdots \frac{2}{2}$ .  
Задача: а) программа, составляющая строку  $S$ , где  $N$ -ий элемент  
цикла - максимальное кол-во леск для каждого шага  $N$ .  
б) модуль полученного кода - то же самое, что  $N$  квадратов переведен в шага  
и скомбинирована строки.

Алгоритм:

1) Предположим, что  $N$  чётное. Тогда  $(N \leq 15000 \text{ и } N \geq 1)$ , ищем  
будет ли  $N$  делиться на 4.

2) Объявляем строку  $S$ ,  $S = "43"$ ; переменную  $i$ ,  $i = 1$ ;  $ans = 0$

3) Установка  $N$ : пока  $N$  делится на 4, то длина строки  $S$  изменяется на 2.

4) Прибавление к строке  $S$  строки  $a$ ,  $a = ("3" + "2" \cdot i) \cdot 2$ ,  
 $i$ -ый кусок размер  $i$ , а каждому шагу убираем  $i$  на 1.

5) Ищем количество строк, фиксированное  $\geq N$ . Это количество 1го шага  $N$  равно?

(одинаково, что первый символ имеет номер 1) прибавляем к ней.

$ans$  значение  $\text{int}(S[1])$ . ( $\text{int}(S)$  первая строка  $S$  имеет)

5) Выберите  $ans$  - ответ на задачу

⊕



N3

*Aerogramme*:

- 7) Объявим  $\varphi$ -но  $H\mathcal{O}\mathcal{Y}(a, b)$ , называемого ногой:  
 нога  $a$  не содержит нуля и  $b \neq 0$ : если  $a > b$ , то  $a = \underline{a:b}$ , иначе  
 $b = b \vee a$ . Значит,  $\varphi$  есть  $\varphi$ -уникальный набор ног - ноги.

±

2) Strobenen a u n na yebel-e yashess zaganu ( $a >= 0$  u  
})

$n > a \wedge n > o \wedge \text{Hog}(a, n) = 1 \wedge a\text{-ganz} \wedge n\text{-ganz}$ , weitere  
werte von a und n

3) m.k     $b \in [0; n)$  u b-gene, no other women nephews bee  
               a-a Number each branchwise generation

(1) Еще к-т = tree, баклажан b, а также баклажан "Не азы карынъем".

12

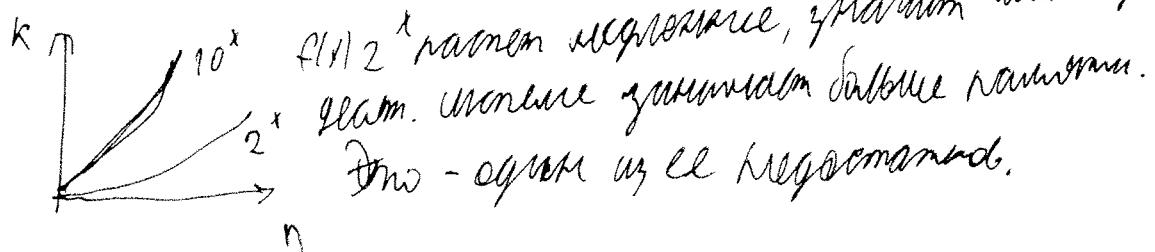
<sup>VR</sup>  
P - turns b gen. armens more numerous. to ~~glossy~~

& 2-well chamber!

$2^{K-1} \leq n < 2^K$  - ~~то~~ нечетные группы в задаче

g-2-20-mon mem - ll:

$$10^{k-1} \leq n < 10^k - k \text{ значеб } \text{ при } g(1) = 10^1$$





Можно доказать это подобным: 62-юной единицей получим пока  
умножая 1000...000, а можно умножив на 76.

Кстати: очень легко перевести число из 10 в 76:   
делаю быстрее чем из 2-хой, т.к. переводят надо не все цифры  
одной, а как другую цифру по отдельности (также степень двойки  
которую нужно вычислить = 4, а не к  $k$ -й степени числа), как  
в двоичной].

(3)

Также, сохраняется тождество при переходе десятичн.  
чисел в 2-юную систему. В двоичной тождество теряет  
смысл, поскольку "умножить" - умножить на 10.

✓6

считаем, что записьется в один столбец по 3 числа. Тогда оно

1) Проблема, что записка такого небольшого размера имеет  
лишние "левые" числа, иначе выведем форму обобщенное:

2) Если мы не знаем как-то начинать, то будем склады-  
ванием не массив, а списк. Создадим список  $\text{arr}$ .

3) добавим в списк. arr массив, содержащий результаты всех шаг.

~~но~~  $A = [a, b]$ , где  $a$  - перв - тое число,  $b$  - последнее. массив, т.к.

4) Рассмотрим  $P$  - это общее кол-во символов перв,  $D$  - общее кол-во символов

5) Продолжим по списку, заполнив шаг прибавляя к  $P$  числа

$a$ , &  $D$  - значение  $b$ .

6) Сравним  $P$  и  $D$ . Если  $P > D$ , то выберем "Перв", если

$D > P$ : выберем "Втор". Если  $P = D$  - "Четв".

(4)



V5.

Лист № 4

1) Напишите алгоритм сортировки, чтобы исп. то же в решении.  
а.к надо быстро отсортировать, исп.-еи алгоритм быстрой сортировки.  
A-массив  
Берем случайный элемент из массива  $A[i]$ , и проверим по массиву слева, так ли найден элемент  $A[j]$ , что  $A[i] \geq A[j]$  и справа-зи-т  $A[g]$ :  $A[i] \leq A[g]$ . Изменяя  $A[j]$  и  $A[g]$  местами. Одна из вставок эд-ланки получим список, в котором „левая“ часть будет иметь  $A[i]$ , а правая - бывшее. Покуривши  
се можно быстро подумав по принципу покурить получим  
аналог. в порядке возрастания списка.

2). Для решения мы будем использовать способ, как в задаче 6, но  
мы можем отформулировать это в помощь алгоритму массива.

3) Создадим массив (экспериментальный), содержащий блоки  
массивов (раз-таким виду  $B \in \mathbb{R}^{(p \times q)} \times \mathbb{R}^{(r \times s)} \times \mathbb{R}^{(t \times u)}$ ), пред-ко проверки  
эх-запроса: ( $N \geq 1$ ,  $N \in \mathbb{N}$ ,  $X$ -цена шах,  $Y$ -запросы), такие блоки сооб. общим  
4) Сформулируем массив (экспериментальный, используя как аргумент для  
сравнения /взаимопроверки обозначения  $A[i]$ ,  $A[j]$ ,  $A[g]$ ) узел списка

(+)

раз-така.

5) Сортируем блоки по порядку все элементов  $S$ .



*Будж дж. монте  
—*

1) Найдем, что мало "стартовых", а не выведенных чисел деление на 9 - чётное

2) Объявим массивы:

$A = [I, V, X, L, C, P, M]^T$  - вектор различных чисел

$B = [1, 5, 10, 50, 100, 500, 1000]^T$  - знаменат.

$C = [a_1, a_2, a_3, \dots, a_n], \forall a_i = 5 \cdot 10^{k+i} // B[i]$

(например для  $n=23$   $C = [8, 1, 2, 0, 0, \dots]$ )

3) Для всех  $i$ , от 1 до 5, скажем 2:

если  $A[i+1] \geq 1$ ; то  $A[i] = 5$  - 6 концов этого числа или оно делится на 5 единиц, кроме и десятков (для которых "пятерок", "нолями" и "четыре пятерки").

4) ~~представим~~  $s = 2$  строка  $S$ ,  $S = [$  *(S<sub>i</sub>)*

5) Считаем от 7 до 1, прибавляем к строке  $A[i-1][i]$  строку  $S$ .

Чтобы складывать надо  $\chi_{10} = V_{\text{раз}}$ ,  $g_{10} = X_{\text{раз}} \text{ и } g_{10}$ .

Если  $C[i] = B[i+1] - 1$ , то будем прибавлять  $S$  к концу строки

$s_i = A[i-1] + A[i]$

6) Выведем  $S$  - представление 6 в десятичной системе

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7112

ОГ 61-88

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

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:

Салават

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 вводится I
2. Если число равно 2, то на экран вводится I  
(т.е. добавляем I к I, I приписываем справа) II
3. Если число равно 3, то на экран вводится II  
(т.е. добавляем справа II к I)
4. Вводим число 5, при этом на экран вводится V
5. При вводе числа 4 отнимаем от V I  
(т.е. симба от V подпишем I)
6. При вводе чисел 6,7,8 добавляем к V справа I, II, III (т.е. вводим 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 вводится 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 получилось введенное число.
9. Вводим число 50, на экране вводится L
10. Вводим число 100, на экране вводится C
11. Вводим число 500, на экране вводится D
12. Вводим число 1000, на экране вводится M.
13. При введении любого числа добавляем или отнимаем от известного начи числа в римской записи столько, чтобы получилось введенное число.  
При этом отнимать и добавлять можно только те числа, записи которых надо известна в римском виде



N 2

Кодирование информации в двоично-десятичной системе счисления числаобразно можно использовать для кодирования очень важной информации, о которой знает определенный круг лиц. Использование двоично-десятичной системы счисления снижает вероятность утечки информации.

N 3.

1. Вводим модуль  $n$
2. Вводим чётное число  $a$
3. Если модуль  $n$  чётное число  $a$  можно разделить без остатка на число большее 1, то умножим его на обратное значение в четвёртой степени. В таком случае на экран выводим "-".
4. Если модуль  $n$  чётное число  $a$  можно разделить только на 1, то выводим число 6.
5. Для нахождения числа  $b$  из значений модуля  $n$  и чётного значения числа  $a$ .
6. Полученное результатом выводим на экран.

N 4

Поскольку квадрат расположено в виде спирали, то сделав небольшой чертёж, можно заметить определённый порядок передования рек. Самое большое число квадратов 15000

3	2	2
2	3	3
2	4	2
3	2	3



из-за перегорания количества реек можно  
найти следующим образом:

$$4 + 2 \cdot 3 + 2 \cdot 3 + 2 \cdot 3 \cdot 2 \cdot 2 + 3 + 2 \cdot 2 \cdot 3 + 2 \cdot 3 + 2 \cdot 3 + 2 \cdot 3 + 2 \cdot 4 + 3 \dots$$

$$\dots + 2 \cdot 125 + 3 + 2 \cdot 123.$$

Поскольку число квадратов не должно превышать 15000.

Число реек равно:

$$\begin{aligned} & 10 + 3 \cdot 849 + 2 \cdot 110 + 2 \cdot 310 + 2 \cdot 510 + 2 \cdot 710 + 2 \cdot 910 + 2 \cdot 1110 + \\ & + 2 \cdot 1310 + 2 \cdot 1510 + 2 \cdot 1710 + 2 \cdot 1910 + 2 \cdot 2110 + 2 \cdot 2310 + \\ & + \cancel{2 \cdot 2510} + \cancel{2 \cdot 248} + \cancel{2 \cdot 244} + \cancel{2 \cdot 246} + \cancel{2 \cdot 248} + \cancel{2 \cdot 125} + \cancel{2 \cdot 123} = \\ & = 33653 \end{aligned}$$

7

Ответ: минимальное число реек - 33653.

v6

1. Вводим имена игроков Петра и Данила
2. Результаты Петра фиксируются в "результат 1", а результаты Данила в "результат 2"
3. Просматривать результаты Петра и Данила на всех страницах в конфигурации
4. Сравнивать полученные данные
5. Если число огней у игрока, чьи результаты фиксировались в "результат 1", то на экран выводится имя Петра
6. Если число огней у игрока, чьи результаты фиксировались в "результат 2" больше, то на экран выводится имя Данила

7

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

412  
0Г 61-55-

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ ГЕЛИВЕРСТОВ

ИМЯ Илья

ОТЧЕСТВО ДМИТРИЕВИЧ.

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

Бычак Ю. Ильин  
Юрьев

program my4;  
var N,r,k,i :integer;  
sum: longint;  
begin  
readln(N);  
if N=1 then writeln(4);  
if N=2 then writeln(7);  
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(sum);  
end;  
end.

(+) (-)



№3

```
program my3;
var a, b, n, a1, n1, f: integer
begin
readln(a, b);
if a > n then writeln('Неправильное значение')
else begin
a1 := a;
n1 := n;
f := 0;
while n1 <> a1 do
  if n1 > a then n1 := n1 - a
  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;
  else writeln('Не существует');
  if f = 0 then writeln('Не существует');
end;
end.
```





№6

Program my6;

var m, N, i, j, sum1, sum2

a:array [1.. 100; 1..3] of integer;

begin

write ln('Ведите количество строк и столбцов матрицы');

readln(m, N);

for j:= 2; sum1:= 0;

for i:= 2 to m do

for j sum1:= sum1+ a[i, j];

j:= 3;

sum2:= 0;

for i:= 2 to m do

sum2:= sum2+ a[i, j];

if sum1 &gt; sum2 then writeln('Первый')

else if sum2 &gt; sum1 then writeln('Второй')

else writeln('Третий');

end.

Как-то настолько равно:

(как-то строк - 1)

Перв-1 строк (sum1)

Втор-2 строк (sum2)

Считаю, что все при  
следующих подсчетах  
увеличиваются в как-то кратном

6

№2

Данная система счисления может использоваться  
 с дробными числами, с числами с плавающей запятой,  
 а также ~~и~~ ~~пере~~ при записи больших чисел где удобства  
 перевода в другие системы счисления. (-)

Так же её применение возможно, если нужно закодировать  
 другие символы помимо цифр, т.к. 6<sup>16</sup> символов не занимает.



N1

you must  
do

program my1;  
var a:array[1..13] of char;

begin

readln(n);

a[1]:='I';

a[2]:='IV';

a[3]:='V';

a[4]:='IX';

a[5]:='X';

(7)

a[6]:='XL';

a[7]:='L';

a[8]:='XC';

a[9]:='C';

a[10]:='CD';

a[11]:='D';

a[12]:='CM';

a[13]:='M';

end;

while n

if (n mod 10) < 3 then begin

X:= n mod 10;

while x>0 do

  s:= s+a[1]; 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»

№ группы

AN 49-36

Вариант №

7102

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)

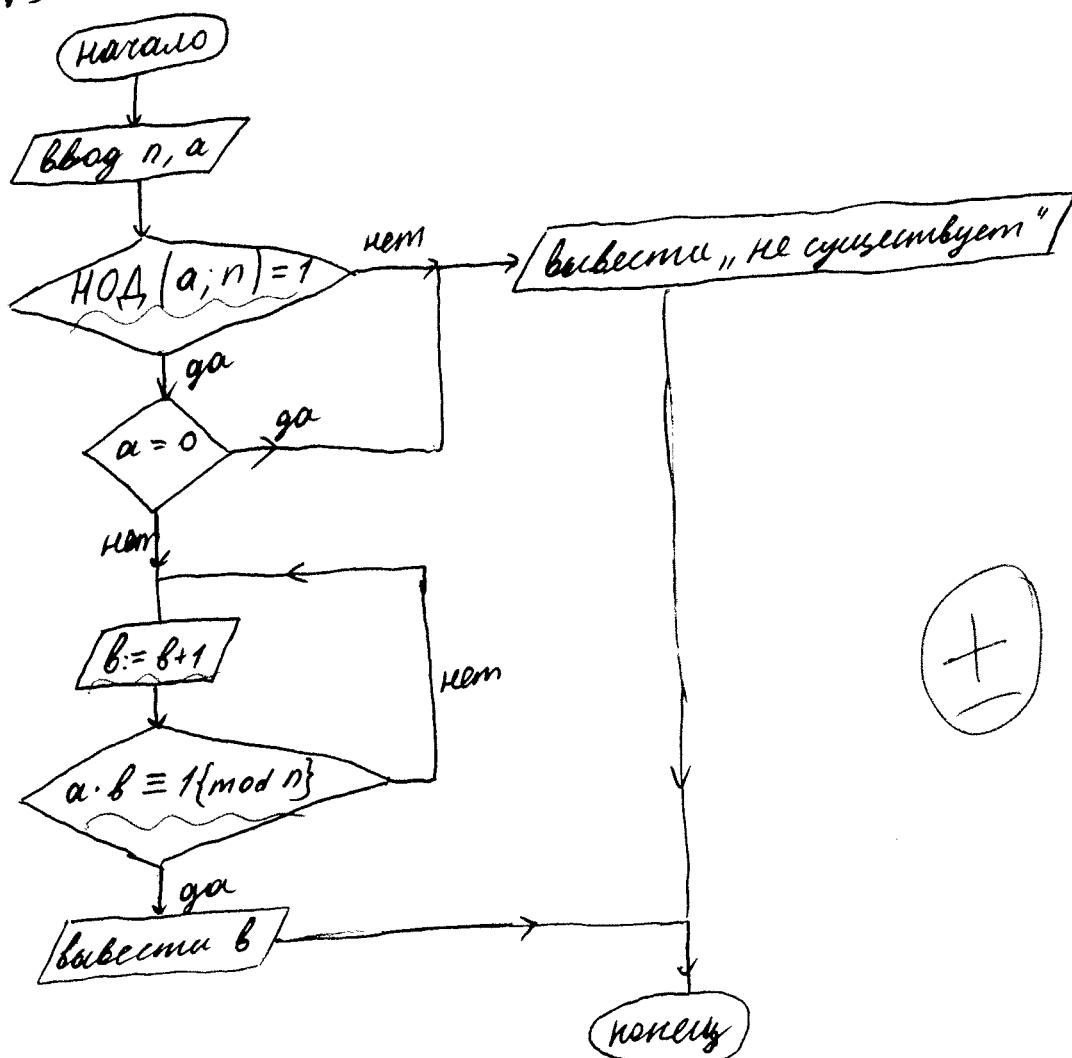
Подпись участника олимпиады:

Edt -

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

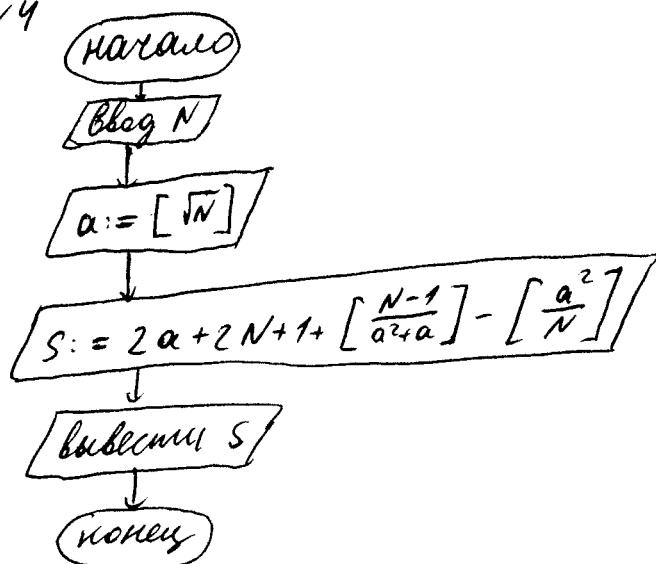


N3



конец

N4





N1

- 1) Слагаемое переведём число  $n_8$  в десятичную систему счисления:

$$\textcircled{1} \quad a = ((m \text{ div } 10^{l-1}) \text{ mod } 10) \cdot 8^{l-1} + ((n \text{ div } 10^{l-2}) \text{ mod } 10) \cdot 8^{l-2} + \dots + ((n \text{ div } 10^0) \text{ mod } 10) \cdot 8^0, \text{ где } l - \text{количество цифр в числе } n.$$

- 2) Так как выражение вместо цифр использовали записи римских букв соответствующие числу  $10^i$  или  $5 \cdot 10^i$ , где  $i \in \mathbb{Z}; j \in \mathbb{Z}; i \geq 0; j \geq 0$ , ~~но~~ мы пусть  $\{a\}$  - запись римской буквы соответствующей числу вида  $10^i$  или  $5 \cdot 10^i$

$$\text{Тогда } 2 \cdot 10^i = \{10^i\} \{10^i\}$$

$$3 \cdot 10^i = \{10^i\} \{10^i\} \{10^i\}$$

$$4 \cdot 10^i = \{10^i\} \{5 \cdot 10^i\}$$

$$5 \cdot 10^i = \{5 \cdot 10^i\}$$

$$6 \cdot 10^i = \{5 \cdot 10^i\} \{10^i\}$$

$$7 \cdot 10^i = \{5 \cdot 10^i\} \{10^i\} \{10^i\}$$

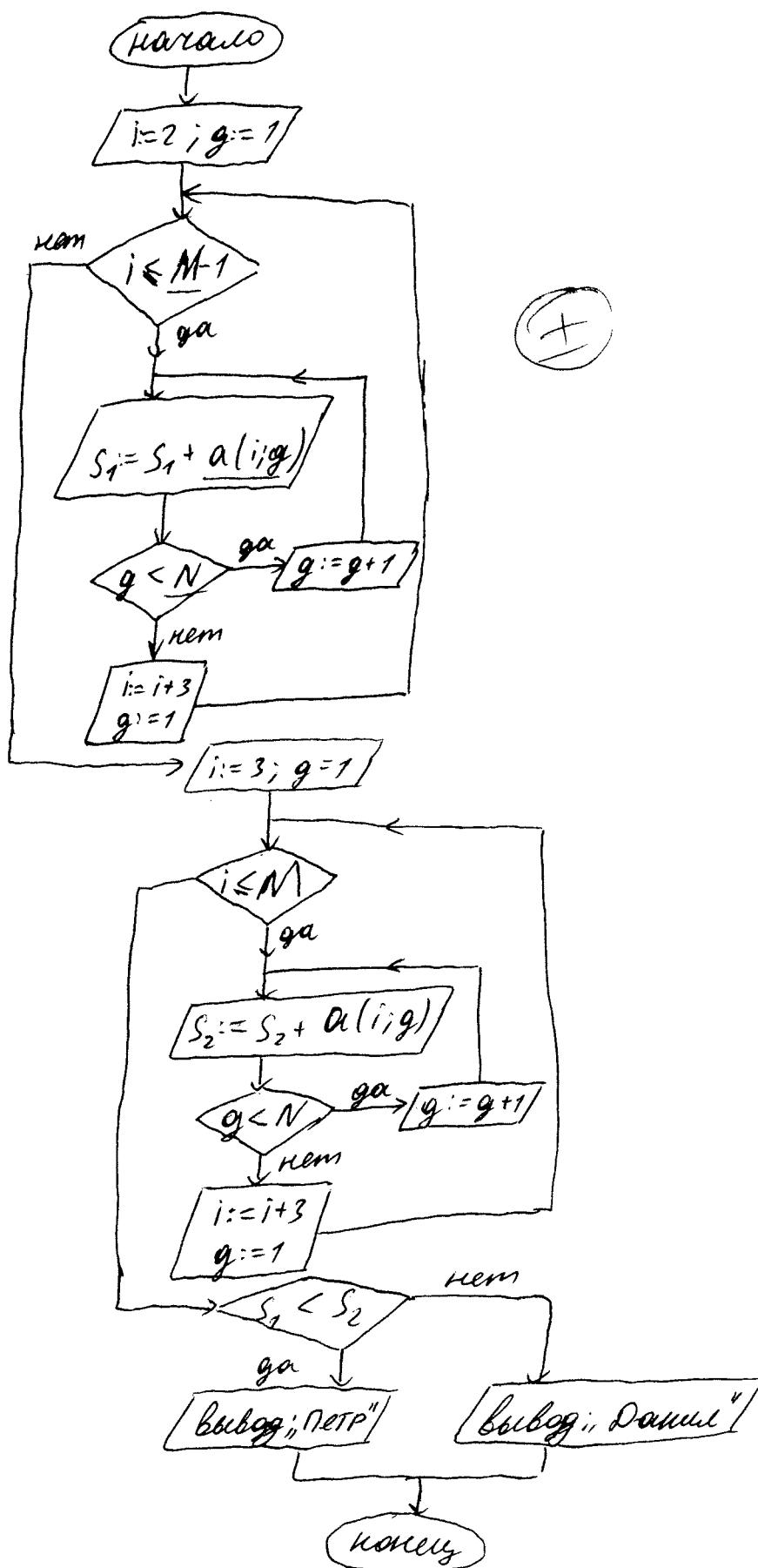
$$8 \cdot 10^i = \{5 \cdot 10^i\} \{10^i\} \{10^i\} \{10^i\}$$

$$9 \cdot 10^i = \{10^i\} \{10^{i+1}\}$$

- 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  $a$ , и умножив  $a_1 \cdot 10^{k-1}; a_2 \cdot 10^{k-2}, \dots, a_k$  на соответствующие составляющие римских букв мы получим ряд записанных букв, соответствующий числу  $n_8$  представленного в виде римских цифр



№6



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7/13

YL 32-30

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ СЕНДЕР  
ИМЯ Анастасия  
ОТЧЕСТВО Леонидовна  
Дата рождения 31.10.1997  
Предмет ИНФОРМАТИКА  
Работа выполнена на 5 листах  
Класс: 11  
Этап: ЗАКЛЮЧИТЕЛЬНЫЙ  
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); { ввод десятичного числа}

f1:=1; f2:=1; { это первые числа Фибоначчи}  
x:=0; y:=0; { величина следующего числа}

s:=''; { строковая переменная для записи в строку в формате с.с.б}

while N &gt; 0 do begin

f1:=1; f2:=1; x:=0; y:=0; { значение первого для каждого

брояза брояза. } { значение первого для каждого брояза брояза}

while ~~f1 < N~~ do begin

x:=f1;

y:=f2;

f1:=f1+f2;

f2:=x;

end;

f1:=f1-y;

{ алгоритм нахождения числа Фибоначчи, которое можно вычесть из данного числа}

N:=N-f1; { разложение данного числа на разности}

~~for i := 1 to N do~~

c1:=0; c2:=0; t:=0; { c1 и c2 - это числа Фибоначчи, предыдущие

t:=c1;

c1:=t;

c2:=t;

d:=0;

if (c1 &lt;= 0) and (c2 &lt;= 0) then begin

while x < c2 do begin

m:=x;

x:=x+y;

d:=d+1;

y:=m;

end;

d:=d-1; { алгоритм поиска конца последовательности чисел Фибоначчи

сновающихся между двумя, которые есть в разном, т.е.

как бы супер. }

N2 - Нет



```
for i:=1 to d do
    S := S + '0'; // Число чисто пустое
    S := S + '1'; // Формируется, приведенная, т.к. из него
                    // есть одно число
end;
end;

t := 0; // Помощник для перенесения используется, чтобы
        // передать получившее в строке число из
        // строки в число

val(t, S, code); // перевод из строки S в число t
write(t); // Проверка
end.
```

N6.

Числоподавление для хранения дается в виде алгоритма  
использованием двоичных переносов — списки. Это называется ~~запоминанием~~  
использованием машинки и отличается от метода  
обработки полученной информации схемореглаждением.

Алгоритм:

объявляем едиичноимущее переменные M, N, K,  
 $X, L, P;$

начало программы

создаем первый список элементов (назовем  $N-M+1$ );  
каждый элемент списка содержит единицами  
изображающую число ярусов башни (первый  
, последний  $-N$ ). Количество единиц  
соответствует числу ярусов,  
указанные на изображающей башне списка.

затем все элементы списка

изображаются

количество единиц определяется числом ярусов  
присвоенное каждому кону,  
которую считают;



создан второй список (кол-во элементов  $\leq 1000/2$  –  
к-во км-в километров, переведены в метры и разделены  
на 2, т.к. она занимает не более  $2 \times 2^2$  позиций)

Для всех элементов списка (кол-во рядов матрицы)  
начало цикла

для всех созданных новых списков элементов  
(кол-во элементов  $\leq 1000/2$ , это самое рис.):

каждый элемент списка содержит дескриптор  
из двух широррессажей: указанный на первом из  
элементов списка, число друзей есть лемок  
коинремнейшии ени;

конец списка;

параллельно прошагиваем оба созданных  
списка;

для всех элементов второго списка  
начало цикла

для всех элементов каждого списка исходного  
списка

начало цикла



Всю широррессажную переделали и представили  
запись, выдающаяся каждому другу лемок ени;

для всех элементов первого списка  
начало цикла  
если и равен нулю друзей лемок, то

коинремни ени увеличиваются на 1;

конец цикла;

конец цикла;

конец цикла.

для всех элементов первого списка  
начало цикла

перебор широррессажей элементов:

число друзей лемок и коинремнейшии

конец цикла;

запись второго списка;

конец программы



N3  
program Z3;  
var p, q, 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; *алгоритм проверяющий числа p-1 близких к*  
 $t := \log(p) \text{div } \log(2)$ ; *{*  $t = \log_2(p)$  *}*  
 $f := 0$ ;  
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  $n := K * 2$ ;  
 $N := 1$ ;  
for l := 1 to K \* d do  $N := n * N$ ;  
if N mod p = -1 do f := 1; *{* проверка второго условия *}*  
end;  
end; *{* f = 1 then write('repository error') else write('working'); *}*  
end.





№5.

Начало программы  
создаем список элементов (т.е. целых, номера и значения  
на месте блоков  $\{ \}$  заменя),  
изначально список пуст.

Назем первое число, присваиваем ее первому элементу списка;  
Назем следующее число, пока не закончена запись  $N$  чисел;  
для всех элементов списка

назаду списка  
проверяется весь список, сравнивается текущий элемент  
с изначальным элементом списка;  
если оно совпадает, то вставляется в начало списка  
новый элемент если оно отличается, вставляется в конец  
списка нового элемента. если нет, или другое условие  
не выполняется, ищут среди списка (убирающийся перебор) и вставляют  
впереди списка;

Назем следующее значение первого числа  
когда списка; Назем передаваемы  $M$  и  $N$ ;  
для всех элементов списка

назаду списка  
выходит в одну строку  $M$  элементов, 5  
переходите к следующей строке;

конец цикла;  
затем список;  
конец программы.

№4

```
program 24;
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»

№ группы

Вариант № 7103

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ Смальков  
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 Фибсс (чел N)  
начало

чел: типы  $\in \mathbb{N}$

тип  $\text{симв} = 1;$

тип  $\text{цел} = 2;$

чел:  $\text{pointer} = 1;$

пока тип  $\text{симв} < N$  делам вложение числа Фибоначчи  
из

$\text{pointer} = \text{pointer} + 1;$

типы  $\text{симв} = \text{симв}[\text{pointer}-1] + \text{симв}[\text{pointer}-2];$

ку

$\text{pointer} = \text{pointer} - 1;$

для  $i = \text{pointer}$  вниз до 0 делам

из

если  $N = \text{типы}[\text{pointer}] \neq 0$  тогда

$N = N - \text{типы}[\text{pointer}];$

~~вывести, 1;~~

~~типы~~

~~иначе, выводим, "0";~~

~~типы~~

конец

ку

N2

алг алг Промежуточки (чел  $N_1$ )  
начало цел  $\text{мин} = 0$ ,  $\text{count} = 0$

чел  $\text{мин} \text{мин} = N_1 + 51;$

для  $r = N_1$  до  $N_1 + 50$  делам

из  $\text{count} = 0;$

для  $i = N_1$  до корень из  $R$  делам

из

для  $j = N_1$  до корень из  $R$  делам

из

для  $k = N_1$  до корень из  $R$  делам

из

для  $l = N_1$  до корень из  $R$  делам

из

если  $i \cdot i + j \cdot j + k \cdot k + l \cdot l = R$  тогда

$\text{count} = \text{count} + 1$

ку

подразумеваем, что  
корень из  $R$  округлен  
или это целое  
число



1/2 (упрощение)

ищ  
ищ  
ищ  
если count < min тогда  
min = count;  
numb = R;

(+)

ищ  
выбираем R мин  
измен.

N3

алг Миллер-Рабина (дел N)

начало

дел t, s, d, a, copy, flag ;

$t = \log_2(p)$  ;

$s = 0$  ;

copy = p-1 ;

пока copy мод 2 == 0 делать  
ищ

copy = copy / 2 ;

$s = s + 1$  ;

ищ

$d = copy$  ;

flag = 0 ;

~~если (a в степени d, под p-1) не равна~~  
~~flag~~ ;

дела i = 1 до t делать  
ищ .



## N 3 (продолжение)

$a =$  случайное от 1 до  $p-1$

$tlay = 0;$

если  $(a^b \text{ степени } d) \bmod p = 1$  тогда

$tlay = 1;$

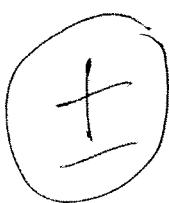
перейти на следующий шаг  
иначе

для  $j = 0$  до  $S-1$  делить

иначе  $a$  в степени  $((2 \text{ в степени } r) \cdot d)$

$tlay = 1$

перейти к следующему шагу



если  $tlay = 0$  тогда вывести „составное“

иначе

если  $tlay = 1$  тогда вывести „вероятно простое“.

конец

функция  $NULL$   $\overset{N^4}{\leftarrow}$   
последовательность целых чисел и возвращающая последовательность

чисел

действие

функция  $NULL$  (дел  $a[i]$ )

делает

$tlay = 1;$

для  $i = 0$  до последний индекс массива

иначе если  $a[i] \neq 0$  тогда

$tlay = 0;$

вывод из цикла

если  $a[i] < 0$  тогда

иначе

если  $tlay = 1$  вернуть 1, иначе 0



## ч 4 (продолжение)

конец  
 // Теперь введем функцию вспомогательную  
 // единица из массива, проверяющую, что передан  
 // сам массив, а  
 // не его конец.  
 // началь  
 функция minus (a[ ])

$$a[0] = a[1];$$

если  $i = 0$  // конечная единица  
 введено из

если  $a[i] == -1$  тогда

$$a[i] = 9;$$

$$a[i+1] = a[i+1] - 1;$$

иначе

// основное тело

алг Динаморифтизма ()  
 начало

дел ~~N[0 ... 1000]~~;  $N[0 ... 1000000]$ ;

~~M[0 ... 1000]~~;  $M[0 ... 1000000]$ ;

~~ROM[0 ... 1000000]~~; ~~M[0 ... 1000]~~; // Max - наибольшее число  
~~ROM[0 ... M]~~;  $ROM[0 ... 1000000]$ ;

вводим подмаково в массив  $N$ , используя  
 число  $N$  так, чтобы первым элементом  
 массива  $N$  являлся крайней правой цифрой,  
 второй является второй с правой цифрой  
 Там же вводим массив  $M$ .

Описываем фигуру элементами обоих массивов  
 обозначаем как 0.

Обозначаем фигуру, ROM, как  $N$ .

Думать  $t = 1000000$

(7)



нома  $\text{NULL}(M) == 0$  деламъ

иу ~~for~~ ~~int~~  $i = 0$  до  $N - 1$  деламъ

иу ~~for~~ ~~int~~  $j = 0$  до  $t$  деламъ

иу  $\text{res}[i + j] = \text{res}[i + j] + \text{prom}[i] \cdot N[j];$

иу

иу

for ~~int~~  $i = 0$  до  $t - 1$  деламъ

иу

$\text{res}[i + 1] = \text{res}[i + 1] + \text{res}[i]$  ~~int~~  $\text{div} 10;$

$\text{res}[i] = \text{res}[i] \text{ } . 10;$

иу

пром присвоимъ деление  $\neq$  res  
res обнулимъ

Minus ( $M[i:j]$ ):

иу

Создать массив res;

конец.

~~Введен фрагмент проверки на отсортированность.~~

~~функция check ( $\alpha \in \mathbb{Z}^N$ )~~

~~иу начало for flag = 0;~~

~~for i = 0 до N деламъ~~

~~иу~~



N5

Выведем логическую формулу, используя  
чтобы ли упорядоченность матрицы

(+) (-)

функции  $\text{check}(\text{a}[N][M])$   
помощью

for  $i = 0$  to  $N-1$  делать

    for  $j = 0$  to  $M-1$  делать

        если  $a[i][j] > a[i][j+1]$

            flag = 0;

            выйти из цикла;

            выйти из цикла;

    иначе

        for  $i = 0$  to  $N-1$  делать

            если  $a[i][M] > a[i+1][0]$  тогда

                flag = 0

        если  $flag == 1$  тогда вернуть 1,  
        иначе 0;

иначе.

алгоритм Сортировка Матрицы (для  $N, M$ )

    для Matrix [ $N][M]$ ;

        для  $i = 0$  до  $N-1$  делать

            для  $j = 0$  до  $M-1$  делать

                если  $b[j] < b[i]$  тогда Matrix [ $i][j$ ];

            иначе

                иначе



пока ~~check ( Matrix < N & I & J ) == 0~~ делаем  
иначе  
    затем  $i = 0$  до  $N - 1$  делаем  
        затем  $j = 0$  до

пока ~~check ( Matrix [ N ] [ I ] [ J ] == 0~~ делаем  
иначе

    затем  $k = 0$  до  $M - 1$  делаем

иначе

    затем  $i = 0$  до  $M - 2$  делаем

иначе

    затем  $j = 0$  до  $M - 1$  делаем

иначе

    если  $\text{Matrix} [ K ] [ J ] < \text{Matrix} [ K ] [ I ]$   
иначе поменять местами  $\text{Matrix} [ K ] [ I ]$  и  $\text{Matrix} [ K ] [ J ]$

иначе

иначе

    затем  $i = 0$  до  $M - 2$  делаем

иначе если  $\text{Matrix} [ i ] [ M ] < \text{Matrix} [ i + 1 ] [ 0 ]$

иначе

    поменять местами  
 $\text{Matrix} [ i ] [ C ]$  и  $\text{Matrix} [ i + 1 ] [ C ]$

иначе

иначе

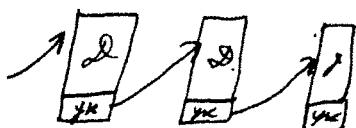
    сформиру Matrix ;

иначе



N6

Воспользуемся числовым списком, т.к  
и задача иначе как-то решается



Для него определим алгоритм  
решения:

- создание списка списка
- добавление элемента в конец
- удаление элемента в начало

Для этого будем список списков,  
чтобы не путать списки учащихся  
или для Пират (для  $N, M; L, K, O$ )

дел список [ $M - N$ ];  $\text{flag} = 0$

заполним этот массив  $O$ ;

будем список результатов  
составлять список и список

когда на ноде есть элементы

ища на ниже ноду есть элементы  
и

ищем на нижних нодах если  
нет

ищем флаг  $= 0$

если флаг равен  $M - N$

ищем i = 1 до указаний  
ищем

если указаний нет заканчиваем

если указаний есть и ищем ноду  
ищем список i; j = список i; j + 1;

и записываем флаг  $= 1$   
заканчиваем запись



1) ~~з добавлением элементов в конец списка и  
этим  $f_{\text{last}} = 0$ , когда  $\text{last} = \text{last}$~~   
~~и у~~

~~и у~~  
~~если~~ Вывески машин состоит  $N - N_3$ ;  
~~и конец.~~

В начале списка ~~з добавлении~~ приведены  
если  $N < N_3$

~~или~~  $N \geq 0$  или  $M < 0$  или  $L < 0$  или  $K < 0$   
~~или~~  $Q < 0$  тогда ~~вывески~~ "Неверно все  
данные введены"

~~При выполнении~~ списка

1) Если  $f_{\text{last}} = 0$  (значит нет ~~записей~~ записей  
и участков) вывод  
одноточие участков в списке -1  
з добавлением нового элемента в список  
из списка ~~результатов~~ не удаляем участков.

Так в массиве содержатся  
записи для ~~всех~~ машин машин,  
и в списке ~~последние~~ конечные и  
еще записи ~~исходящие~~ на конец.

*выдан дополнительный лист*

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7092

*ZE 57-25*

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ Соловьев СОЛОВЬЕВ  
ИМЯ Сергей СЕРГЕЙ  
ОТЧЕСТВО Андреевич АНДРЕЕВИЧ

Дата  
рождения 20.09.1999

Класс: 9

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 4 листах

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

Подпись участника олимпиады:

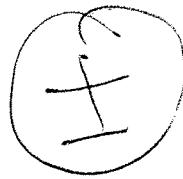
Впишите свою фамилию имя и отчество печатными буквами, дату рождения, класс, название предмета, общее количество листов, на которых выполнена работа и дату выполнения работы.



н 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.
```



н 4

Программа на языке PASCAL:

```
VAR N, M: longint;  
i: integer;  
BEGIN  
KREAD (N, M);  
FOR i := 2 to M do  
BEGIN  
N := N * N;  
VAR N, N1, M: longint;  
i: integer;  
BEGIN  
READ (N, M);  
N1 := X 1;  
FOR i := 1 to M do  
BEGIN  
N1 := N1 * N;  
END;  
WRITELN (N1);  
END.
```





二

1

Психологическое значение синтеза — сложное значение, в котором „личность“ может быть идентифицирована с тема, связанная с её биографией/жизнью в турбулентном разрыве: тем более что синтез напоминает о турбулентном разрыве, тем меньше её „личность“, и наоборот, тем больше синтеза от турбулентного разрыва, тем больше её „личность“.

Рассмотрим сие доказательство наименее чисто 291.

Чудора, 3<sup>й</sup> имеет разные виды, и в них гравюра бывает, не разной, то есть чудора 1<sup>й</sup>, которая называема враз-  
мер бриллианта.

Планы боевых действий с противником утверждены приказом  
Генерального штаба.

$$\begin{aligned} & \stackrel{3 \cdot 3 \cdot 3}{F} = 3 \cdot 4! + 3 \cdot 3! + 3 \cdot 2! + 3 \cdot 1! = 72 + 18 + 6 + 3 = \\ & = 99_{10} \end{aligned}$$

Таким образом, "чтобы" убедить (в нашем случае, "з"  
стремление и погоду) в том, что "з", погоду  
в разные годы, неизменного бывает, то "з",  
погоду в разные годы сущна (сравните:  $72 >> 3$ )  $\Rightarrow$   
 $\Rightarrow$  именно ощущение погоды можно считать погодой.

Ambon; ja, ebneval.

| n 3

## Программа на языке PASCAL:

VAR  
N, N1, A, K: INTEGER;

## BEGIN

READ(N);

$N := N$

$$K := \emptyset.$$

R. - 5

WHILE ( $n1 < 2^* n$ ) DO

(Am. Na. At.-emp.)



```

BEGIN
A := N1 mod N;
IF (N div A < > 0) THEN K := K + 1;
N1 := N1 + 1;
END;
WRITELN (K);
END.

```

n 2

Программа на языке PASCAL:

```

VAR
i, GT, T : INTEGER;
G: ARRAY[1..27] OF INTEGER;
G[1]:= -12
G[2]:= -11
G[3]:= -10
G[4]:= -9
G[5]:= -8
G[6]:= -7
G[7]:= -6
G[8]:= -5
G[9]:= -4
G[10]:= -3
G[11]:= -2
G[12]:= -1
G[13]:= 0
G[14]:= 1
G[15]:= 2
G[16]:= 3
G[17]:= 4
G[18]:= 5
G[19]:= 6
G[20]:= 7
G[21]:= 8
G[22]:= 9
G[23]:= 10
G[24]:= 11
G[25]:= 12
G[26]:= 13
G[27]:= 14

```

BEGIN

WRITE ('Номер заселого поселка: ');

Read (i);

WRITE ('Добавиме энтульское брюхе', ', час

READ (GT);

T := GT + G[i];

WRITELN [T];

n 4

VAR

K, A, n, i: real;

b: array [1..1000] of real;

(ин. сир. штамп)



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&gt;0) do

begin

if (n div b[k]=4) then write(x,'\_'); //,\_ - пробел //

n:=n mod b[k];

k:=k-1;

end;

end.



## Олимпиада школьников «Надежда энергетики»

СШ 59-89

← Не заполнять  
Заполняется  
ответственным  
работником

№ группы

Вариант №

7102

шифр

ФАМИЛИЯ

Сорокин

ИМЯ

Иван

ОТЧЕСТВО

Эдуардович

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:

Иван

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.

бюф  $M$ ;бюф  $N$ ;перемешиваем  $\text{petr} = 0$ ; // сумма результатов Петра. $\text{dan} = 0$ ; // сумма результатов Дании. $h$ ; // высота таблицы результатов $l$ ; // ширина таблицы результатов. $i$ ; // строка изменения массива.если  $M > N$ , то  $\text{petr} = h = M$ , иначе  $h = N$ ; // узнаем как большее  $P = M$ ; // узнаем как большее табличка результатовесли  $h < 2$  или  $l < 3$ , то выбор "Неправильные размеры таблицы," иначепрерываем таблицу результатов и будем обрабатывать массив  $a[i][j]$ ; $i = 2 \dots h$ 

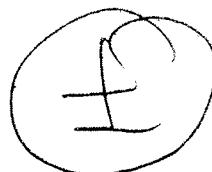
Н. Ч.

 $\text{petr} = \text{petr} + a[i, 2];$  $\text{dan} = \text{dan} + a[i, 3];$ 

К. Ч.

если  $\text{petr} = \text{dan}$ , то выбор "Ничья";если  $\text{petr} > \text{dan}$ , то выбор "Петр";если  $\text{petr} < \text{dan}$ , то выбор "Дани";

К. А.



№2.

Это будет числоворазно, если нам нужно не выгородить с боями членов этого класса.

Проект программы не так будет выглядеть иначе, ибо будет работать именно с боями самого класса и сменами преобразованием только членов. А в двоично-десятичной коде мы можем преобразовывать такие цифры, менять их местами, заменять и т.д.

№3 - Лист



N4.

H.K.

беск  $N$ ,

нечётное

 $S=10$ ; // сумма рек $Sch=4$ ; // четное количество $p=1$ ; // кол-во рек между трине.  
 $t, i$ ; // где нетораесли  $N < 1$  или  $N > 15000$ , то беск "Нечётное значение  $N$ ".

иначе

если  $N=1$ , то беск "4",

иначе,

если  $N=2$ , то беск "7",

иначе

если  $N=3$ , то беск "10",

иначе

H.U. Пока  $Sch \leq n$ 

H.U.

 $i = 1 \text{ до } 2$ 

H.U.

 $t = 1 \text{ до } p$ 

H.U.

 $S = S + 2$  $Sch = Sch + 2$ если  $Sch = n + 1$ , то остановить все циклы;

K.U.

 $S = S + 3$  $Sch = Sch + 1$ если  $Sch = n + 1$ , то остановить все циклы;

K.U.

 $P = P + 1$ .

K.U.

беск  $S$ ;

K.U.

K.A.





N5.

Н.А. Высокая таблица ряда максимума ее значение № n/n.

$$h = \max \sqrt[n]{n/n}; f=2;$$

Возьмем для обнуления массивов:  $a[i, 3]$  - исходная таблица редуцируется и  $b[f, 3]$  - наращивающая таблица в ходе алгоритма.

~~f=2~~ по h пока массив  $a[i, 3]$  не будет нулевым

Н.У.

$$i = 2 \text{ по } h$$

Н.У.

проверить все элементы  $a[i, 3]$  и нахождение maxa

К.У.

$$i = 2 \text{ по } h.$$

Н.У.

если  $a[i, 3] = \max a$

Н.У.

$$b[f, 1] = t; b[f, 2] = a[i, 2]; b[f, 3] = a[i, 3]$$

К.У.

$$f = f + 1;$$

Удалить строку ряда maxa, и убрать из массива  $a[i, 3]$  строку из трех ненулевых строк.

К.У.

если нулево, то втор. массив  $b[h, 3]$

К.А.

N1.

Н.А.

ввод n; p=|n|; s=0; f=0;

while  $p \neq 0$

$$q = p \bmod 10$$

$$s = s + q \cdot 8^f;$$

$$p = p \cdot \operatorname{div} 10;$$

$$f = f + 1.$$

К.У. если  $n > 0$ , то  $n = p$ , если  $n < 0$ , то  $n = -p$ .

Это переведен число  $n$  из десятичной системы в восьмеричную.

Теперь выбираем для числа  $n$  соответствующее число в восьмеричной системе счисления.

К.А.

(-)

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 712

74 25-69

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ

Стольная

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:

Анна

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 : int64; a:text;
begin
  assign (a, in.txt)
  reset (a);
  K:=1
  while not eor(a) do begin
    read (a, mas[K]);
    inc (x)
  end;
  x:=2;
  y:=3;
  repeat
    begin
      n:=n+mas(x);
      K:=K+3;
      z:=z+mas(y);
      y:=y+3
    end;
    until (x>length mas) or (y>length mas)
    if n>z then writeln ('Нимр побегул');
    if z>n then writeln ('Дарин побегул');
    if z=n then writeln ('Жарса');
  end.
```

(5) ~~5~~

5.

Хранение - таблица . идёт в строке x : 1-N<sup>o</sup>; 2 - значение ; 3 - результат.  
Алгоритм:

1. Проконденсация таблицы с начала до конца . Для этого 2-ух соседних строк проверка - если в строке x значение [x;3] > значение [x+1;3], тогда:  
2. Стока y = строка x ([x;1], [x;2] и [x;3] совпадают . присв. к [y;1],  
[y;2]; [y;3])

(1)

3. Заменить строки x присваиваемся соотв. знач - я и - об из [x+1].  
4. Заменить [x+1] присв. соотв. знач - я из y

5. Если не произведено перестановки на шагах 1-и? Если да, то блок-код 1-и еще раз, если нет - засчитают упорядочими по возрастанию.



алгоритм завершён.

2.

Числообразно применять при записи больших чисел; при необходимости быстрого последующего перевода в 10-тичную систему счисления при необходимости передать последовательность цифр, не выкладывая числом.

7

4.

Количество реек, необходимое для окраинного квадрата, является ~~записи~~, а весь числовой ряд имеет форму  $4 + 3 + (3 + 2 \cdot k)$ , где  $k$  увеличивается на 1 после каждого 2-ух повторений цепочки  $3 + 2 \cdot k$ .

Таким образом, иск. число =  $(3 \cdot (N:2)) + (2 \cdot N \cdot 2)$

7

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

SG94-29

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ СУХОВА

ИМЯ ДАЛЬГА

ОТЧЕСТВО ЦГОРЕВНА

Дата  
рождения 09.04.1958.

Класс: 11

Предмет информатика

Этап: заключительный (очный)

Работа выполнена на 5 листах

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

Подпись участника олимпиады:

Сухова

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.

Способ 1: табличный:

Найдем такое число фибоначчи,  $\leftarrow$  пусть оно  $K$ , его номер  $\leftarrow$  порядковый  $n$ .

чтобы оно было больше нашего, но предыдущее (перед ним) было

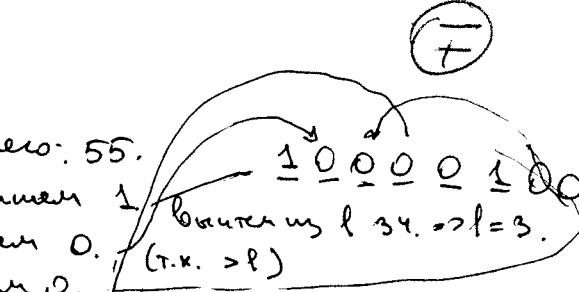
меньше нашего. ~~Запишем 1.~~ Теперь мы знаем, что длина нашего числа  $n$ .  
(в табл. чисел.)Вычитем из  $f - K^{n+1}$ . Теперь перебираем по порядку числа фибоначчи от  $n=2$  до 1. Если  $f \leq K$   $\leftarrow$  числовой, то пишем 0. Если  $f > K$ , то пишем 1 и вычитаем из  $f$  числ. Так пока ~~число~~. Полученная ~~строка~~ посл. из 0 и 1 есть искомое число.

числовой  
номер.  
 $\leftarrow [n]^{n+1}$  порядковый номер.

$K$

→ 37.

первое числ. больше него: 55.

50 него идет 34.  $\Rightarrow$  запишем 190 34 идет 21.  $\Rightarrow$  пишем 0.90 21 идет 13.  $\Rightarrow$  пишем 0.90 13 идет 8.  $\Rightarrow$  пишем 0.90 8 идет 5.  $\Rightarrow$  пишем 090 5 идет 3.  $\Rightarrow$  пишем 1. (т.к.  $\leq l$ )  $l = f - k^{L+1}$ , где  $i=3$  т.к. 3-порядковый член 3.90 3 идет 2  $\Rightarrow$  пишем 0 } т.к.  $l=2$ ,90 2 идет 1  $\Rightarrow$  пишем 0.(т.к.  $> l$ )

② Как искать число фибо?

Получил 10000100.

Создаем массив так, что  $k^{L+1} = 1$   $k^{L+2} = 1$ .и дальнейшие числа:  $k^{L+i+2} = k^{L+i+1} + k^{L+i+2}$ пока  $k^{L+i+2} < l$  это повторяется.если  $k^{L+i+2} > l$ , то  $i=n$ . и начинаем раскладывать.

£



§ 2.

Сделаем "умный" перебор чисел от  $N_1$  до  $N_1+50$ .

$$\min = -1000; f_1 = 0.$$

Пусть  $i$ -числа от  $N_1$  до  $N_1+50$ . (по очереди)

$P \geq 0$ . ( $P$  делает работы между)

Пусть  $j$ -числа от 1 до  $\lceil \sqrt{i} \rceil$

Пусть  $k$ -числа от 1 до  $\lfloor \sqrt{i-j^2} \rfloor$

Пусть  $f$ -числа от 1 до  $\lfloor \sqrt{i-j^2-k^2} \rfloor$

чтобы  
случайно  
не  $\rightarrow$   
участок  
несколько  
раз одинако-  
вое назначение

Если  $i = j^2 + k^2 + l^2 + f^2$  то:

если  $j > k > l > f$  или  $j > k > l = f$  или  $j > k = l > f$  или  $j > k = l = f$  или

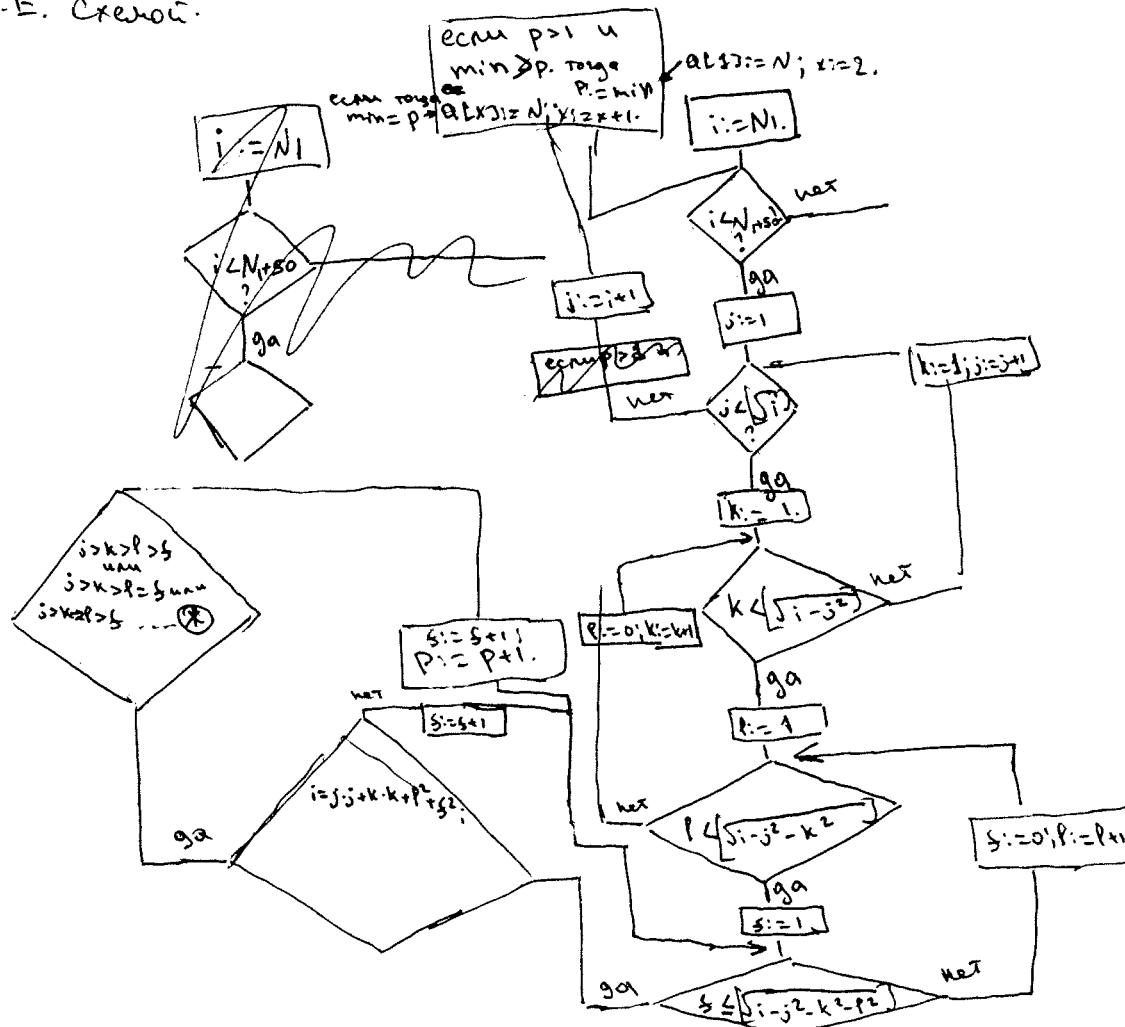
$j = k > l > f$  или  $j = k > l = f$  или  $j = k = l > f$  или  $j = k = l = f$  тогда

$P$  увеличивается на 1.

Прекращаем все переборы кроме  $i$  (заканчиваем перебор  $jklf$ ).

Смотрим, если  $P > 1$  и  $P \geq \min$  тогда  $\min = P$ .

Т.е. Схемой.







54.

(хорошо одно)

~~Если длина числа 0, то выводят.~~1 Запишем число  $\overset{N}{\overbrace{\dots}}$  массив и перевернем.т.е.  $105 \xrightarrow{\text{int}} \underline{501}$ с  $M$ -аналогично.Затем создадим двумерный массив, в котором  $N \times N$  строк $M_1 = N$ . $M$  раз генерим следующее:2. Создаем  $a$  такой, что он  $M_1 \times N$  - строкдублирует  $M$ , то все строкив  $i$ -ую строчку умножаем на  $i$  элемент массива  $N$ .

(каждое её элемент).

раскидываем так, чтобы в 1 элементе было 1 чифра.

т.е.

$$\begin{array}{r} 18 \quad 32 \quad 6 \\ \hline 21 \quad 26 \end{array}$$

$$\begin{array}{r} 18 \quad 32 \quad 6 \\ \hline 8 \quad 9 \quad 3 \end{array}$$

$$T.k \quad 18 \bmod 10 = 8.$$

$$18 \div 10 = 1. \Rightarrow 32 + 1.$$

$$\text{т.е. } 32.6 \Rightarrow 9.$$

$$ac[i] = a[i] \oplus (10 + q[i]).$$

$$a[i] = ac[i] \bmod 10.$$

Внимательно смотрим, если выходит за пределы - это

далее 1 элемент сдвигает на ~~плюс~~<sup>0</sup> вправо,2 раз. на ~~плюс~~<sup>1</sup> 1N раз на ~~плюс~~<sup>0</sup> N-1.

получим:

$$\begin{array}{r} 22 \\ \hline 21^3 \end{array} \quad 12 \quad \begin{array}{l} 2 \text{ раза} \\ \downarrow \end{array}$$

$$\begin{array}{r} 12 \\ \hline 12 \end{array} \rightarrow \begin{array}{r} 12 \\ \hline 24 \end{array} \rightarrow \begin{array}{r} 12 \\ \hline 24 \\ 24 \end{array} \rightarrow \begin{array}{r} 12 \\ \hline 024 \end{array}$$

далее сложение

$$\begin{array}{r} a_1, a_2, a_3 \\ b_1, b_2 \end{array}$$

$$a_1 + b_1, a_2 + b_2, a_3$$

Складываем все элементы по столбцам. раскладываем  
так, чтобы в 1 элементе 1 чифра. наибольшее значение  $M$  раз. не меняется (наибольшее)



№6.

Собираем джипы, сколько есть - какого вида.

Составляем массив из элементов<sup>†</sup>, размера такого, сколько существует.

\* - эти элементы из 3х чисел. 1- сколько грузов есть  
2- ее место по ОХ  
3- ее место по ОУ.

Если какое-то ею меняется или умирает, то

из массива удаляется ~~ее~~ и все сдвигаются на 1.

(или если хранить с помощью одинаковой, то  
указатели предыдущего скидываются на следующий)

Если вырастает новое бешка - меняется по  
рё координате число грузов.



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

713**ST81-96**

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ

Тимофеев

ИМЯ

Валерий

ОТЧЕСТВО

ВитальевичДата  
рождения23.10.1997Класс: 11

Предмет

информатикаЭтап: заключительный

Работа выполнена на

4 листах

Дата выполнения работы:

15.03.2015  
(число, месяц, год)

Подпись участника олимпиады:

Валерий

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 Введем число, например 37. Разложим его на слагаемые так, что эти самые слагаемые будут являться числами Фибоначчи. Так 37 такими числами будут являться 34 и 3. 34 нет смысле раскладывать дальше, равно как 3, т.к. где единицы рядом стоять не могут. Необходимо рассматривать именно из 34, 3. Компактное выражение множества, условий или для: 1 - эти числа не являются последовательными в ряду Фибоначчи, и 2 - эти числа принадлежат ряду Фибоначчи. Запомнив эти слагаемые продолжим по цепочке, шаговая цепочка, задавая условие выхода, где ~~не~~ конечное число счетчиков равно предыдущему номеру самого золотого слагаемого. Например, условием выхода из нашего числа будет являться такие: когда счетчик "N" примет порядковый номер числа 34, т.е., когда  $N=8$ . По мере шага каждого числа проверяется если на этом предыдущем номере счетчика нашего слагаемого, то пишем в строку выхода "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 "1", "0", что и будет являться числом Фибоначчи.

③ Введем число  $p$ .

Представим число  $p-1$  в виде  $p-1 = 2^s \cdot d$ , где  $d$  - нечетное число. Тогда согласно предыдущему условию  $a = 0 + 1 \cdot 0(p-1)$ , где  $a = 0$ .



Например: если первенца счетчик равна 1, то проверка 1,  
 если 2, то 2; и т.д. до ( $p-1$ ). Пусть  $a$  - первенца счетчик.  
 Проверки по следующим условиям:

Если  $a^d \equiv 1 \pmod{p^3}$ , тогда

если

существует цикл ~~с одинаковыми~~, пусть счетчик в этом  
 цикле =  $t$ , тогда цикл от 0 до ( $S-1$ ) с предустановкой, где  
~~если~~ ( $q=0$ ) ~~или~~ ( $q \leq S-1$ ), то выполняется цикл  
 в котором каждое разумение имеет значение  
 применения  $q$  от 0 до ( $S-1$ ) до тех пор, пока  
 условие не становится правильным, а именно:

$$\left( a^{(2^t)} \right) \left( a^{1/(2^t) \cdot d} \right) = -1 \pmod{p^3}.$$

(На самом деле эти уравнения можно  
 было начинать обозначить условие выхода из цикла,  
 т.е., как только оно становилось правильным, что приводило  
 к выходу из цикла.) Однако, если уравнение превратилось  
 вправда в  $q=1$ . Выход из цикла

Если  $q=1$ , то просто продолжается первое разумение  
 в начале каждого блока цикла, т.е. каждого повторения.

Если ~~это~~ после выполнения цикла основного,  $q$  остается  
 равным 1, то число вероятно простое, иначе число составное.

Конечная проверка: Если ( $q=1$ ), то выходит  
 вероятно простое, иначе - составное.

Однако, я не уверен о том, какую деталь.

Все проверки Ниллерса-Рабинса это проверки наложенной  
 для + случайно выбираемая  $a$ . Тогда могут правильные.



Проверки будут члены обнаружаться тоже, однако сами члены будут от 1 до  $t$  или же другое условие  $q=0$ . где  $t$ -это  $\log_2(p)$ . члены  $a$  будут выбирать случайно от 1 до  $p-1$ .

- ④ Вычислим  $N^M$ , где  $N$  и  $M$  могут быть от 1000 цифр.

Естественно, что мы один разок программировали и способен однажды вычислить только ограниченные числа, однако для этого существует ограничение арифметики, не пополучим членами, а получим остатки ~~или~~ или членами, однако запись в конец ~~записи~~ приведет к нарастающим в новом числе.

7

- ⑤ Мы имеем массив, размером  $M$  на  $N$  клеток. Создадим двумерный массив размерностью  $M$  строк и  $N$  столбцов. Построим проверка по двумерному массиву менять ряды стоящие числа, элементы массива, возникающие от этого величины, если два соседних не равны, а  $\oplus$  постепенно добавляя предыдущего, то менять их местами, иначе оставлять как есть место. Такой сортировки называется "пузырек". Самый простой и примитивный,



## №5 (продолжение)

Зато действенный способ. Таким образом

максимальный элемент оказывается на первом месте и весь массив будет отсортирован.

⑥ Создаем двумерный массив  $K$  на  $N \times N$ , тогда если есть узелка не более  $2m^2$ . Переводим все в меры, балы получаем распределение шата т.е. Стартуя, продвигаясь по строкам массива и на каждом квадрате строки проверим „если“. Входит в цикл от  $M \text{ до } N$ . Каждую строку (н повторение) числа проверяет:

Если есть ~~бала~~, то инициализации переменную ~~F~~ счетчик. Как только появится кол-во ~~бала~~ ~~уровней~~ ~~бала~~ инициализации ~~тот~~ ~~счетчик~~ инициализации, которая отвечает именно ~~за~~ если с этим кол-вом ~~уровней~~. Таким образом „продвигаясь“ по всему массиву-помощи. В конце просто подводим итоги. На выходе имеем переоценки и подчищаем сильно уровней.

⑦ Создаем число  $a = (N+1) \text{ до } (N+50+1)$  на исходном числе проверяем число  $N$  (которое изначально равно  $N+1$ ), которое наворачиваем число  $N$  увеличивается на 1 до  $N = N+50$ . раскладываем число  $N$  всеми возможными способами на спасенные и проверяем обработали ли они издевательства, если да, то инициализации счетчик  $d$ . Выходили на экран, после выполнения ~~и~~ счетчик  $d$ . Он и показывает кол-во таких цифр для промежутка от  $(N+1 \text{ до } N+50)$ .

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7103

04 30 - 79

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ Тихонова  
ИМЯ Варвара  
ОТЧЕСТВО Андреевна  
Дата рождения 15.09.1998  
Предмет информатика  
Работа выполнена на 5 листах  
Класс: 10  
Этап: заключительный  
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  $\text{fib}(n)$ , возвращающая число Рибонacci под порядковым номером  $n$ . Результат будет храниться в переменной  $S$ , где инициализировано ненулевым числом, которое будет длиной ~~меньше~~ исходного числа. (Будем пока  $\text{fib}(m)$  инициализировать равным 1 ( $m=1$ ), меньше длины числа  $X$ , увеличиваем  $m$  на 1. Затем (после выхода из цикла) вычитаем 1 из  $m$ .  $S$  содержит  $m$  '0'.)

Затем, пока  $X$  не равен 0, будем выполняем цикл (пока  $\text{fib}(m)/\text{fib}(m-1) < k$ ,  $m++$ ), вычитаем 1 из  $m$ .

из  $k$  вычитаем  $\text{fib}(m)$ , заменяя на место  $m$  в строке  $S$  вставляем '1'.

Код на C++:

~~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[m] = '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 == n) return (cur);~~

~~}~~



№5.

Сначала нужно ~~вставить~~ ~~вставить~~ все числа в список, а затем отсортировать его по возрастанию. Рассмотрим алгоритм сортировки пузырьком.  $\text{tmp}$  - переменная с текущим списком;  $a$  - список, из которого нужно сортировать;  $\text{list}$  - список

Представим  $\text{list}$  как двумерный массив символов  $[M; N]$ .

Нужно в цикле проходить по всем элементам списка, забывая о последнем, если tmp[i] < tmp[i+1], менять местами tmp[i] и tmp[i+1].

Список  $a$  - неотсортированный список всех чисел с  $\text{list}$ .

В цикле  $i \rightarrow n-1$  проходить по всем элементам списка ~~записи~~ по блочному циклу  $j \rightarrow n$  проходить по элементам с  $i+1$  до последнего, сравнивая  $a[i+j]$  и  $a[j]$ . Если второй элемент первого меняет их местами.

Выход получаем отсортированный список чисел  $a$  с  $\text{list}$ .



№ 11.

Найдите прямое сопротивление между M<sub>1</sub> и J<sub>1</sub> и замените  
бесконечное сопротивление каждого промежуточного  
сопротивления M<sub>1</sub>. Однако, если сопротивление M<sub>1</sub>  
простое или его прямое сопротивление бесконеч-  
ное, то переходите к прямому сопротивлению M<sub>2</sub>  
перемычки № 6 и начните эти же промежуточные  
изменения.

\* Однако, так как дано только сопро-  
тивление по параллелю N и M, ~~и~~  
переходите к параллелю по времени и  
по нему организуйте по времени и  
памяти, что можно просто

Следите от 180 M до кон-  
ца пути перебора (он изображено  
равен 1) по N, но дальше сопротивле-  
ние)

сравнивайте по времени)

Наконец найдите для каждого  
каждому не превосходящего его сопротив-  
ления замените найденное сопротивление и  
запишите в блок навигации в строку 2, и  
будут получены в строке 1.

перемычки № 6 сопротивлений 2<sup>1</sup>, M<sub>1</sub>-K<sub>1</sub>  
где находящееся выше в сопротивлении буде 2<sup>2</sup>  
следующими блоками с участием  
замены на сопротивление к раз-



2. Нужно  $f(n)$ - функция, возвращающая True, если  $n$  можно представить в виде суммы ~~четырех~~ квадратов натуральных чисел.

В ней хранится некоторый ~~некоторый~~ массив  $f[0..n]$  таких чисел, сама функция  $f$  содержит проверку в цикле  $i$  числа находящегося перед  $f$  в массиве  $f$ . Каждый из четырех квадратов числа, каждым из которых проверка конечных чисел на равенство и проверка того, является ли их сумма числом  $n$ .

Проверка на равенство может и не числами и возвращаем True, если не все они равны между собой.

Затем находятся все ~~числа~~ и создаётся массив  $N$  из ~~максимум~~ ~~минимум~~ ~~максимум~~ ~~минимум~~ чисел с  $k$  по  $n+50$ .

Затем для каждого из чисел  $N$  и их квадратов будут проверять от  $k$  числа в аксовом массиве.

Выводим те числа, значение которых получено из квадратов минимум.



№ 6.

Решите лучше всего хранение в таблице вида:

N = параметр	1	2	3	4	5	...	Q
X							
Y							
числ							
числ?							

+ (	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">E; L3 км</td></tr> <tr><td style="padding: 2px;">координаты</td></tr> <tr><td style="padding: 2px;">E; k3 км</td></tr> </table>	E; L3 км	координаты	E; k3 км	
E; L3 км					
координаты					
E; k3 км					
- (	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">код-во строк N; 1</td></tr> <tr><td style="padding: 2px;">есть ли значение</td></tr> <tr><td style="padding: 2px;">единица (0/1)</td></tr> <tr><td style="padding: 2px;">единица (0/1) sa</td></tr> </table>	код-во строк N; 1	есть ли значение	единица (0/1)	единица (0/1) sa
код-во строк N; 1					
есть ли значение					
единица (0/1)					
единица (0/1) sa					

при подсчете строка содержит список чисел и  
число строк от N до M и состоящем  
из чисел списка, где "строка" =  $\{ \cdot \}$  и  
"числ?" =  $= \cdot \cdot \cdot$ .  
Также для упрощение работы с  
таблицей можно удалить элементы  
последней позиции которой = 0.  
№ 3 столбца наименований S. Их можно заменить  
числом Важнее пока d не станет неравним  
нулю приводим d значение  $(P-1) : 2^S$ , где  
устанавливаем S. Но бывает из чистых усло-  
вий S на единицу.

Далее находим  $\log_2 P$  и если  $d = 1$  и  
проверяем не все члены. Для этого храним  
1 переменную - flag, значение которой изменяется  
на false, если членов не хватило. Если  
 $flag = \text{false}$ , значит из программы, выведен  
данные остаток. Для проверки бывают  
несколько способов в списке  $\{ 0; 1; 2; \dots; 7 \}$  и проверяем  
числовое значение всех P. Если где-то верно,  
то члены прошли. Выводим, что число  
вероятно правильное.

## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7102**СВ 59-11**

шифр

← Не заполнять  
Заполняется  
ответственным  
работником

ФАМИЛИЯ ЩЕВЪЁВАИМЯ НАДЕЖДАОТЧЕСТВО СЕРГЕЕВНАДата рождения 18.08.1998 Класс: 10Предмет Информатика Этап: ЗаключительныйРабота выполнена на 5 листах Дата выполнения работы: 27.02.15  
(число, месяц, год)Подпись участника олимпиады: Щевёва

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

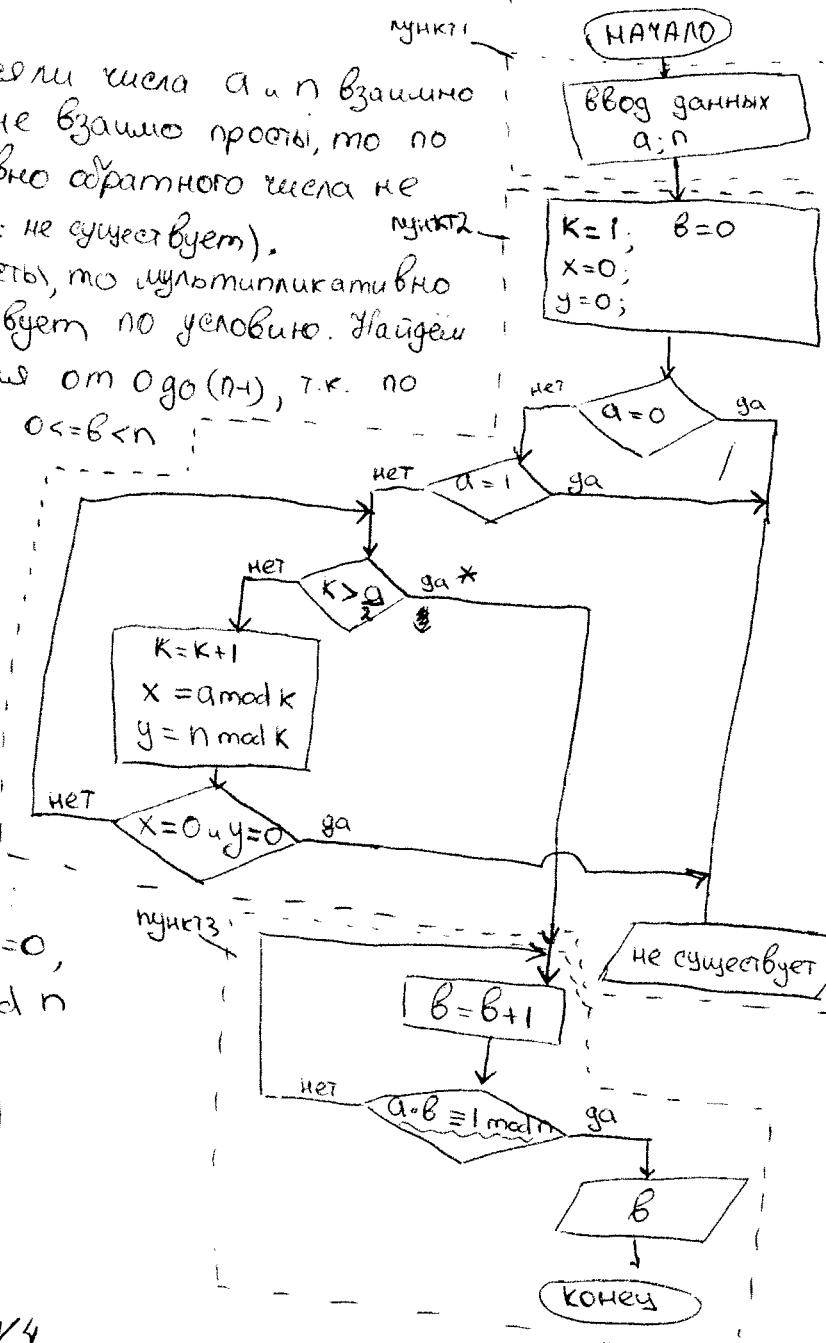


№3

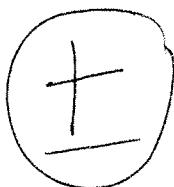
найди доп. числа к найденным  
Уснер Уснер

1) Введите данные (в качестве примера приведу алгоритмическую 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. Найдём его, проверяя все значения от  $0$  до  $(n-1)$ , т.к. по условию  $\exists b, \text{т.к. } (a \cdot b) \equiv 1 \pmod{n}$  и  $0 \leq b < n$   
 $\Rightarrow$  взаимно просты  $a$  и  $n$

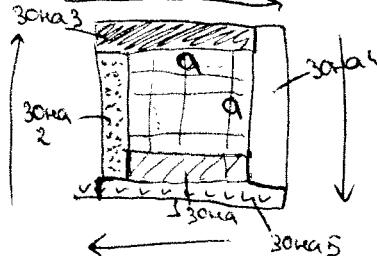


комментарий к пункту 3:  
 $B \neq 0$ , т.к. иначе  $a \cdot B = 0$ ,  
 и значит  $a \cdot B \equiv 0 \pmod{n}$



№4

Заметим, что для  $k$ -ого квадрата нужно 3 рейки, если он граничит с одни и узлы изменяющиеся, и 2 рейки, если он граничит с двумя узлами изменяющимися квадратами. Тогда, если узлы изменяется квадрат  $a \times a$  квадратов, то (сторона квадрата  $(zg \geq a)$  по стрелкам):



- в зонах 2 и 3 "a" квадратами нужно 2 рейки, а одному (первому) нужно 3 рейки

- в зонах 4 и 5 ("a+1" квадратами нужно 2 рейки, а одному (первому) нужно 3 рейки для последующее си продолжение на лице  $OZ$ )

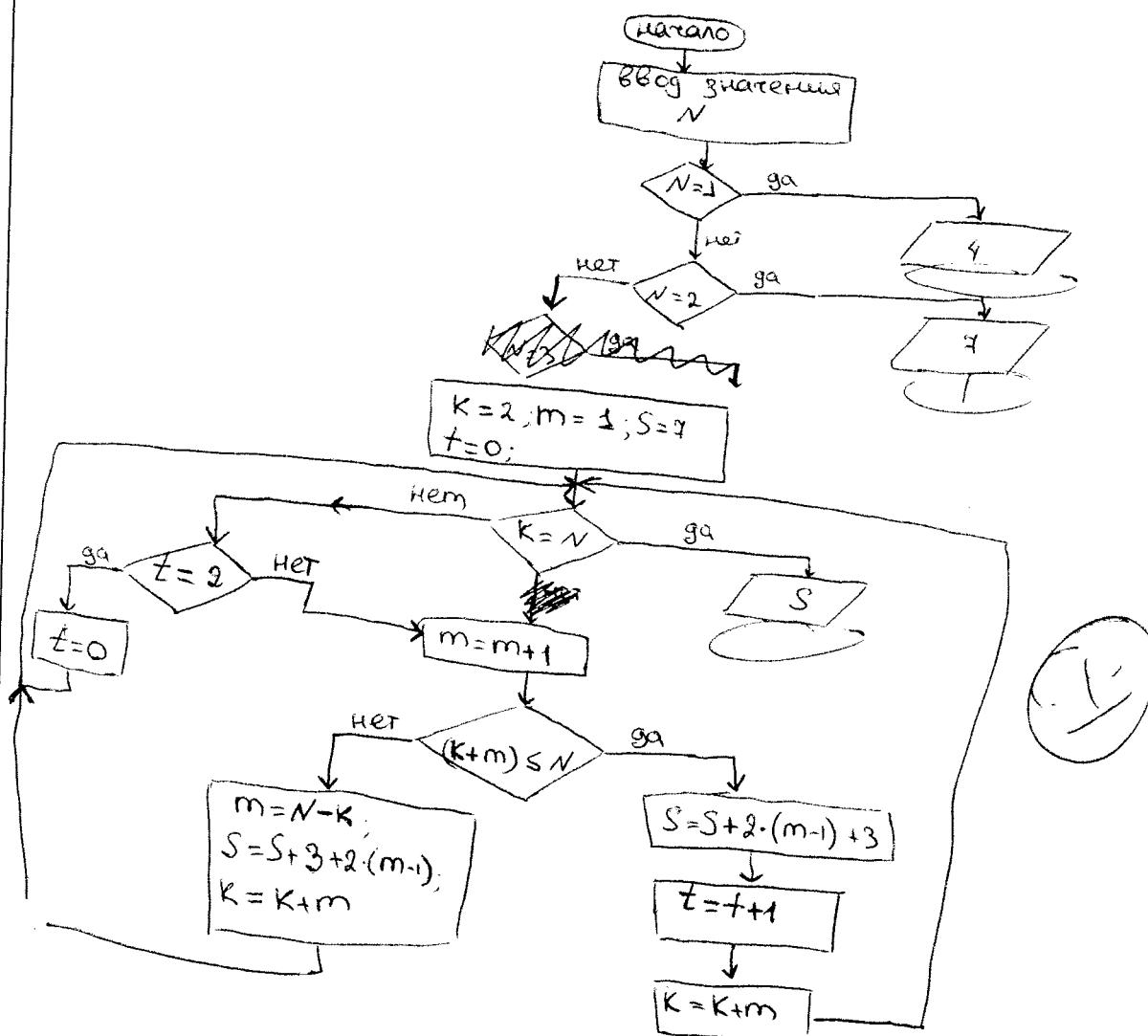


## №4 (продолжение)

Тогда заметим последовательность кол-ва нужных реек для каждого следующего квадрата:

$$4; 3; 3; 2; 3; 2; 3; 2; 2; 3; 2; 2; \underbrace{3; 2; 2; 2}_{3; 3 \times 2}; \underbrace{3; 2; 2; 2}_{3; 3 \times 2}; \underbrace{3; 2; 2; 2}_{3; 4 \times 2} \dots$$

Исходя из этого привожу пример алгоритма, считающего нужное кол-во реек



№2

Двоично-десятичный код удобно использовать, когда ~~запись чисел~~ нужно работать с десятичными числами, записанными в системе счисления типа римской, ~~какие-то~~ ~~одинаковые~~ ~~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)

... т.к. 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 можно выполнить алгоритмические действия)

Двоично-десятичную с.с. удобно использовать, если необходимо знать кол-во символов в десятичной записи.

## N1

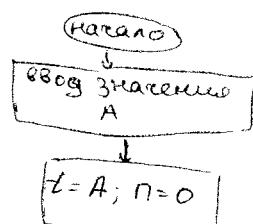
I перевод из восьмиричной с.с. в десятичную с.с.

1) посчитать количество знаков в записи числа A.

2) посчитать  $t$  и  $n$  равно пример:

число A в десятичной с.с. (B):

$$B = (A \bmod 10^0) * 8^0 + (A \bmod 10^1) * 8^1 + \dots + (A \bmod 10^n) * 8^{n-1}$$



II перевод из десятичной с.с. в восьмиричную с.с.

3)  $x_1 = B \bmod 10^3$

4)  $str_1 = \underbrace{M M \dots M}_{x_1}$

5)  $x_2 = (B - B \bmod 10^3) \bmod 10^2$

6) •  $x_2 < 4$ , тогда  $str_2 = \underbrace{C \dots C}_{x_2}$

~~•  $x_2 < 8$ , тогда  $str_2 =$~~

•  $x_2 = 4$ , тогда  $str_2 = CD$

•  $x_2 = 5$ , тогда  $str_2 = D$

•  $5 < x_2 < 9$ , тогда  $str_2 = \underbrace{D \dots D}_{x_2-5}$

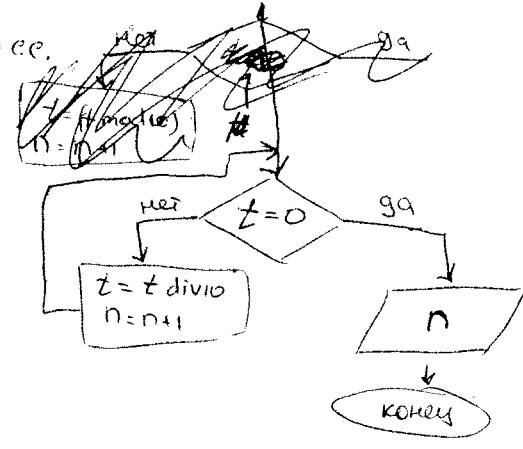
•  $x_2 = 9$ , тогда  $str_2 = CM$

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  $str_3 = \underbrace{L \dots L}_{x_3-5} X$



см. продолжение на стр 04



N1 (продолжение)

8)  $\lambda_3 = 3$ , тогда  $str_3 = XC$

9)  ~~$x_4 \in \{B, D\}$~~ 

$$x_4 = B \bmod 10$$

10)  $\cdot x_4 < 4$ , тогда  $str_4 = \underbrace{I \dots I}_{x_4}$  •  $x_4 = 4$ , тогда  $str_4 = \overline{IV}$

•  $4 < x_4 \leq 8$ , тогда  $str_4 = \underbrace{V \dots I}_{x_4 - 5}$

•  $x_4 = 9$ , тогда  $str_4 = IX$

11)  $str = str_1 + str_2 + str_3 + str_4$

12) выбор ответа: str.

Будем док. метод №  
Установка

Результаты хранить в двух таблицах.

Алгоритм решения задачи

1) в табл. 3. третьей столбец ("результат") упорядочить по  
убыванию значений строк  $n$

2) для каждого значения  $A$  третьего столбца по порядку (сверху  
вниз) проделать действие (в Excel, например, можно написать  
формулу):

1) в таблице два найти значение в  
третьем столбце, равное значению  $A$  (не исключая начальных с  
первой строки. В случае нахождения, остальные строки не  
рассматриваются). 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) значение строке  $K$  второй таблицы  
убрать (не удаляя строку). Чтобы в случае равных результатов  
избежать повторения строк.

NB

В решении задачи будем считать, что данные записываются  
в строку же.

Например 

N1		РЕЗ-71		РЕЗ-72		N2		РЕЗ-71		РЕЗ-72	...
----	--	--------	--	--------	--	----	--	--------	--	--------	-----

См. продолжение на лице 05



№ 6 (продолжение)

Будан дол. Иван № 2

Ученик

I проверить все ли данные помещаются, если кол-во сыгранных партий  $X$

- 1) если  $3X > M \cdot N$ , то не все данные есть в таблице ⇒  
 ⇒ ответить на вопрос не получится

II посчитать кол-во

Рассмотрим несколько вариантов ( $M$ -кол-во столбов).

a)  $M \equiv 0 \pmod{3}$

~~2)~~ II Найдём сумму ожков, набранных Петром

2) просуммируем значение в первых столбцах  $i$ , где  $i \equiv 2 \pmod{3}$ ; результат  $S_1$

III Найдём сумму ожков, набранных Данилом.

3) просуммируем значение в столбцах  $j$ , где  $j \equiv 0 \pmod{3}$   
 Результат  $S_2$ .

b)  $M \equiv 1 \pmod{3}$

II Найдём сумму ожков  $S_1$ , набранных Петром

~~2)~~ для строк 1; 2; 3 и 4; 5; 6 и 7; 8; 9 ...  $N-2; N-1; N$ :

2) Найдём сумму

$$S_1 = (1; 2) + (1; 5) + \dots + (1; M-2) + (2; 1) + (2; 4) + (2; 7) + \dots + (2; M) + \\ + (3; 2) + (3; 5) + \dots + (3; M-1)$$

III Найдём сумму ожков  $S_2$ , набранных Данилом

$$3) S_2 = (1; 3) + (1; 6) + \dots + (1; M-1) + (2; 2) + (2; 5) + \dots + (2; M-2) + \\ + (3; 1) + (3; 4) + \dots + (3; M)$$

c)  $M \equiv 2 \pmod{3}$

аналогично пункту б) разобьём таблицу на таблицы ~~и~~  $M \times 9$

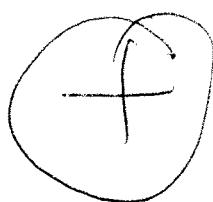
$$II S_1 = (1; 2) + (1; 5) + \dots + (1; M) + (2; 3) + (2; 6) + \dots + (2; M-2) + \\ + (3; 1) + (3; 4) + \dots + (3; M-1)$$

$$III S_2 = (1; 3) + (1; 6) + \dots + (1; M-2) + (2; 1) + (2; 4) + \dots + (2; M-1) +$$

$$+ (3; 2) + (3; 5) + \dots + (3; M)$$

IV Сравнение  $S_1$  и  $S_2$

- $S_1 > S_2 \Rightarrow$  ответ Данил
- $S_2 > S_1 \Rightarrow$  ответ Петя
- $S_1 = S_2 \Rightarrow$  ответ некто



## Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7092ZE 51-28

← Не заполнять  
Заполняется  
ответственным  
работником

шифр

ФАМИЛИЯ ЯРДДЕЕВАИМЯ ЮлияОТЧЕСТВО ЕвгеньевнаДата рождения 28.06.1999Класс: 9 БПредмет информатикаЭтап: ЗаключительныйРабота выполнена на 3 листахДата выполнения работы: 27.02.2015  
(число, месяц, год)Подпись участника олимпиады: Юдит

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 для решения этой задачи, это снабдить число  $p$ . Задача нахождения остатков об делении на  $p$ , будем считать простым с  $p$ . Далее выясним какими числами.

1. Снабдим число  $p$  (на языке программирования Паскаль `read(p)`)
2. Заменим, что ~~если~~ <sup>остаток</sup> при делении числа( $a$ ) на число( $b$ ) меньшее чем  $b$  и <sup>остаток</sup> при делении числа( $a$ ) на число( $b$ ) ~~равен~~ <sup>равен</sup> ~~число на которое делится~~. Значит мы можем <sup>расширять</sup> простых числа от 1 до  $p$  (не включая  $p$ ), безо других возможных остатков. Так не может.

Если число от 1 до  $p$  является простым и  $p$ , то это и есть один из <sup>единичных</sup> остатков при делении на  $p$ , будем считать простых с  $p$ .

Значит нам достаточно <sup>учитывая</sup> простых чисел 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  $k$  подсчитывает количество остатков.

5). Нам нужно найти все возможные варианты индексов, т.е. числа ~~бесконечное~~ больше 9.999 и меньше 100.000.000 (но числа последовательности индексов содержат от 5 до 8 знаков) и делиться на 7 и 11. Но 7 и 11 это взаимно простые числа, значит число ~~известное~~ делится на 7 и 11, делится или 77 ( $11 \cdot 7 = 77$ ). Первым <sup>(последним)</sup> числом будет 10010. Следующее число будет Больше предыдущего на 77.





1. Находим наименчее число удовлетворяющее условию.  
Дав  $\Rightarrow$  10.000 единиц на 77. Получим произведение  
130. Далее  $\overset{5 \text{ (наименчее)}}{\underset{\text{значение числа}}{\ast}}$  77 на 130 и получим 10010. (1)

2. Будем выводить на  $\ast$ ое последующее число прибавляя  
к предыдущему 77, пока ~~не~~  $\Rightarrow$  число не станет  
Больше или равно ( $\geq$ ) 9.999.999.

На языке программирования Паскаль это будет выглядеть так:  
 $b := 9999$   
 while ( $k < 99.999.999^{22}$ ) do  
 begin  
 $b := b + 77$   
 write(b)  
 end;

Мы пишем ~~когда~~ ( $k < 99.999.999^{22}$ ) б.к.

Если мы прибавим к 99.999.999 77 то получим  
99.999.999, и условие будет соблюдено.

(2) Я считаю что самой удобной <sup>способ</sup> определение данных  
о часовом поясах образа, 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). ~~База будет состоять из таблицы~~  
~~База содержит следующие данные: часовые~~  
Будут занесены списки всех стран. (б.к. нам будет удобно  
часовой пояс, указывая его в таблице не нужно).

~~когда появляется утро, мы~~

При определении местного времени, первое что мы  
делаем, это определим утро, ~~и~~ ~~ночью~~ спомощью некоторого  
некоторого заданную страну. Если она находится в таблице  
+1, то мы считаем местное время, прибавив к часам  
по Гринвичу то число, что стоит после GMT в ~~затем~~ часовом поясе.



Если же сортировка переводит время, то имена в временных  
символах прибавляются к часам по правилу числовое-  
вное после GMT в единицах часов после и  
в зависимости от ~~буквы~~ даты сегодняшней прибавляем  
или убавляем 1 час.

④ 1. Считываем имена m и n (read(m; n))

2. Выделяем из m раз:  $k = k \cdot N$ ; где k это число которое  
подсчитывает  $N^m$ .

3. Выводим k. (write(k)).

Т.к. имена получаются дробью времени при сортировке  
программы в Pascalе, необходимо ~~есть longint~~ записать  
 $N$  и  $k$ , как longint. ~~longint не сработал~~

⑤ 1. Посчитаем сумму разрядов чч!

т.е. на языке Pascal это будет выглядеть так:

$k := 1$   
for i = 1 to 44 do

$k := k \cdot i;$

где k подсчитывает значение чч!

2. Запишем то число или строку. a (string).

3. Проверим каждую телесную то есть строку,  
и если она равна "ч", то вывести ~~число~~ число, равное  
длине этой строки минус переднюю позицию ~~значения~~  
значения этой строки минус переднюю позицию ~~значения~~

символа, равного 'ч', и прибавить 1.

т.е. if  $a[i] = 'ч'$  then  $\begin{array}{l} l := \text{length}(a) - i + 1 \\ \text{writeln}(l) \end{array}$

⑥ Я считаю, что для сортировки имена является  
подсчетом байт, т.к. ~~все~~ разряды числа не имеют  
своих значений.

⊕