

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7111

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ БОЛУШЕВА

ИМЯ СВЕТЛАНА

ОТЧЕСТВО ОЛЕГОВНА

Дата рождения 14.09.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

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

Подпись участника олимпиады:

Болушева

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.

CLS

sRome \$ = ""

10 Input "Введите натуральное (целое, большее нуля) число: " nDec

IF nDec > 3999 THEN

Print "Извините, римская система не может записать такое большое число"

GOTO 10

END IF

IF nDec < 1 THEN

Print "ВНИМАТЕЛЬНЕЕ, при вводе! Число должно быть БОЛЬШЕ нуля!"

GOTO 10

END IF

WHILE nDec > 0

IF nDec >= 1000 THEN

sRome \$ = sRome \$ + "M" : nDec = nDec - 1000

ELSE

IF nDec >= 500 THEN

sRome \$ = sRome \$ + "D" : nDec = nDec - 500

~~ELSE~~
ELSE

IF nDec >= 400 THEN

sRome \$ = sRome \$ + "CD" : nDec = nDec - 400

ELSE

IF nDec >= 100 THEN

sRome \$ = sRome \$ + "C" : nDec = nDec - 100

ELSE

IF nDec >= 90 THEN

sRome \$ = sRome \$ + "XC" : nDec = nDec - 90

ELSE

IF nDec >= 40 THEN

sRome \$ = sRome \$ + "XL" : nDec = nDec - 40

ELSE

IF nDec >= 10 THEN

sRome \$ = sRome \$ + "X" : nDec = nDec - 10

ELSE

IF nDec >= 5 THEN

sRome \$ = sRome \$ + "V" : nDec = nDec - 5

ELSE

IF nDec >= 4 THEN

sRome \$ = sRome \$ + "IV" : nDec = nDec - 4

ELSE

sRome \$ = sRome \$ + "I" : nDec = nDec - 1

END IF

END IF

END IF

END IF

END IF

END IF

END IF

END IF

END IF

END IF

WEND

Print "Преобразованное число" sRome \$



N 2.

Система пишется там, где
нужно часть десятичного ввода-вывода
на индикатора (часы, приборы и
тому подобное.) Также упрощено
умножение, деление на 10 и округление.

N 3.

```

Program u;
label 1233
var
  x, otv: word
  k, v, tr: byte.

begin
  Readlnlnl; if n=1 then begin
    kv:=1
    while kv·kv < N do kv:=kv+1
    kv:=kv-1
    tr:=kv·2-3
    x:=2
    otv:=7
    while tr <> 1 do begin
      otv:=otv+6+x·2
      x:=x+2
      tr:=tr-2
    end
    otv:=otv+3+x
    N:=N-kv·kv:
    if n > 0 then
      if w <= kv then begin
        otv:=otv-3+(v-1)
        v:=v-1
      end
      ELSE begin
        otv:=otv+6+1N-21.
      end
  123: write(otv)
  END.

```



№6.

Нужно создать программу, которая будет суммировать результаты 2-х партий игроков, в последующем сравнивать полученные результаты.

Введем переменные, $a; b; c; d$ - 4 результата 2-х игроков.

$a; b$ - 1го игрока

$c; d$ - 2го игрока.

2) $x = \text{Петр}$ сумма $a + b = x$ (результат Петра)
 $y = \text{Данил}$ сумма $c + d = y$ (результат Данила)

3) Получили результаты.

4) Если $x > y$, то Петр набрал больше очков.

Если $y > x$, то Данил набрал больше очков.

5) ~~В ответ~~ ~~задав~~ Ответ выводим игрока, который набрал большее кол-во очков.

6) В качестве результата выведем имя игрока

```
a = InputBox ("Введите результат 1, первого игрока")
b = InputBox ("Введите результат 2, первого игрока")
c = InputBox ("Введите результат 1, второго игрока")
d = InputBox ("Введите результат 2, второго игрока")
IF x = Summ(a+b)
y = Summ(c+d)
```

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7111

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ

ДЗЮБА

ИМЯ

АНТОН

ОТЧЕСТВО

Владимирович

Дата

рождения

15.01.1997

Класс:

11

Предмет

Информатика

Этап:

2

Работа выполнена на 2 листах

Дата выполнения работы:

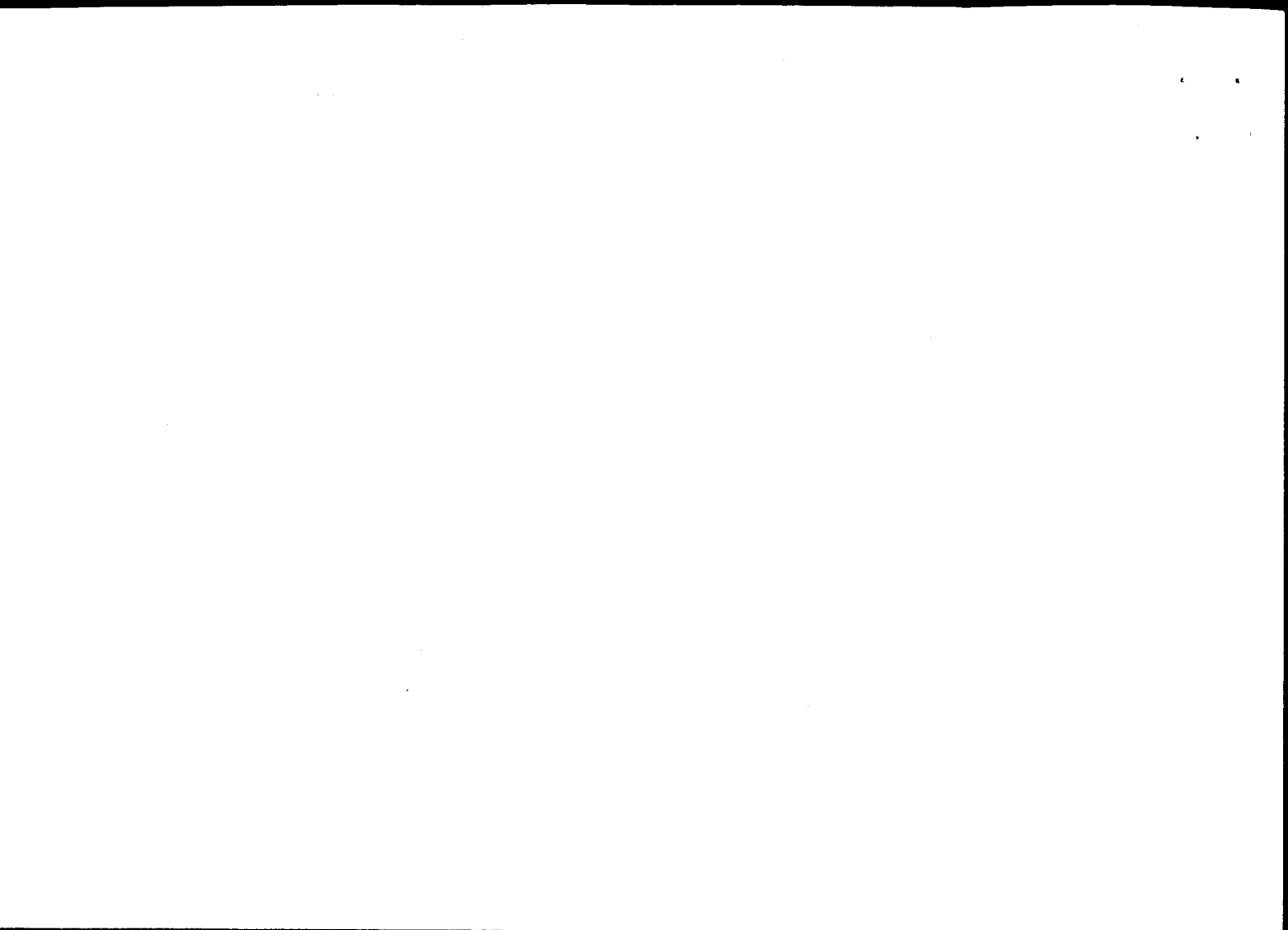
04.01.2015

(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.





Задача 6

Пусть M-высота блока, тогда N-ширина страницы блока.
Пусть rez-т1 - это результат Петра, тогда rez-т2 - это результат Рашида. ЯП-Паскаль.

```

program zadacha 6;
var petr: integer; danil; integer; m: integer; n: integer; e: integer;
a: array [1..1000][1..1000] of integer; d: integer; correct: integer
begin
  {начало программы}
  readln(m); readln(n); {введем M и N - высоту и ширину блока соответственно}
  petr:=0; danil:=0; {предварительно обнуляем переменные, которые будут}
  e:=1; {содержать очки игроков}
  repeat {мы делаем указатель строки единичным по условию}
  petr:=petr+a[e; 2]; {очень, условием выхода из которого будет}
  danil:=danil+a[e; 3]; {является нулевое значение (или отрицательное) номера}
  e:=e+1; {партии.}
  until a[e; 1]=0; {всичке прибавляет к очкам игроков те очки, которые}
  if petr > danil then write('Петр'); {соответствуют данному e (указателю строки}
  if danil > petr then write('Рашид'); {номеру партии)}.
  if danil = petr then write('Ничья'); {выводим результат игры.}
end. {конец программы}

```

```

d:=1;
write('Укажите количество партий в игре');
readln(correct); correct:=correct+1;
a[correct; 1]:=0;
repeat
  read(a[d; 2]);
  read(a[d; 3]);
  d:=d+1;
until d=correct;

```

Задача 5

Пусть результаты написаны на бумаге, тогда в программу должен быть введен механизм сбора данных. Можно составить одну программу, удовлетворяющую всем задачам, поставленным передо мной.

```

program zadacha 2;
var
  id: array [1..10000] of integer;
  b: array [1..10000] of string;
  res: array [1..10000] of integer;
  a: integer; i: integer; k: integer; L: integer; s: string;
begin
  write('Сколько было экспериментов?');
  read(a);
  for i=1 to a do begin
    id[i]=a;
    write('Введите значение параметра', i, ' эксперимента');
    read(b[i]);
    write('Введите результат', i, ' эксперимента');
    readln(res[i]);
  end;
  for k:=1 to a do
    for L:=1 to a do begin
      if (res[L] < res[a]) then begin
        L:=res[L]; res[L]:=res[a]; res[a]:=i;
      end;
    end;

```

~~res~~ s := b[l]; b[l] := b[a]; b[a] := s
 i := id[l]; id[l] := id[a]; id[a] := i; } отсортировали все массивы относительно возрастания значений массива с результатами

алгоритм
 затиши и
 хранит
 данные в
 текстовой
 форме

```

assign (output, 'output.txt'); rewrite (output);
assign (input, 'input.txt'); reset (input);

for i := 1 to a do begin
  writeln ('номер эксперимента:', id[i], 'Значение параметра:', b[i], 'Результат:', res[i]);
  writeln;
end.

```

Задача 2

Применяем такой системный вызов чаще всего целесообразно в тех случаях, где есть необходимость процедуры ввода и вывода десятичных значений. Например, ~~с помощью устройства~~ с помощью устройства калькулятора.

Задача 1

```

Program zadacha_1; uses crt;
var a: array [1..13] of string; b: array [1..13] of integer;
i, k, n: integer;
c, ch: string;
begin
  c := 'MCMDCCLXXCLXXIXVIVI';
  for i := 1 to 13 do begin
    if i mod 2 < 0 then k := 1 else k := 2
    a[i] := copy(c, i, k)
    case i of
      1: b[i] := 1000; 2: b[i] := 900;
      3: b[i] := 500; 4: b[i] := 400;
      5: b[i] := 100; 6: b[i] := 90;
      7: b[i] := 50;
      8: b[i] := 40;
      9: b[i] := 10;
      10: b[i] := 9;
      11: b[i] := 5;
      12: b[i] := 4;
      13: b[i] := 1;
    end;
  end;
  writeln ('Арабское число')
  readln (n);
  i := 0;
  repeat
    inc(i);
  until (n >= b[i]) do
  begin
    n := n - b[i];
    ch := ch + a[i];
  end;
  until n = 0;
  writeln ('Римское число, ch);
end.

```


Задача 3

програм задачи 3

var a, b: integer } a и b - переменные, которые будут содержать входные
 i: longint; c: longint; } данные - модуль и целое число соответственно.
 cor: integer; c1: integer;

begin
 readln(a); } вводем данные
 readln(b);

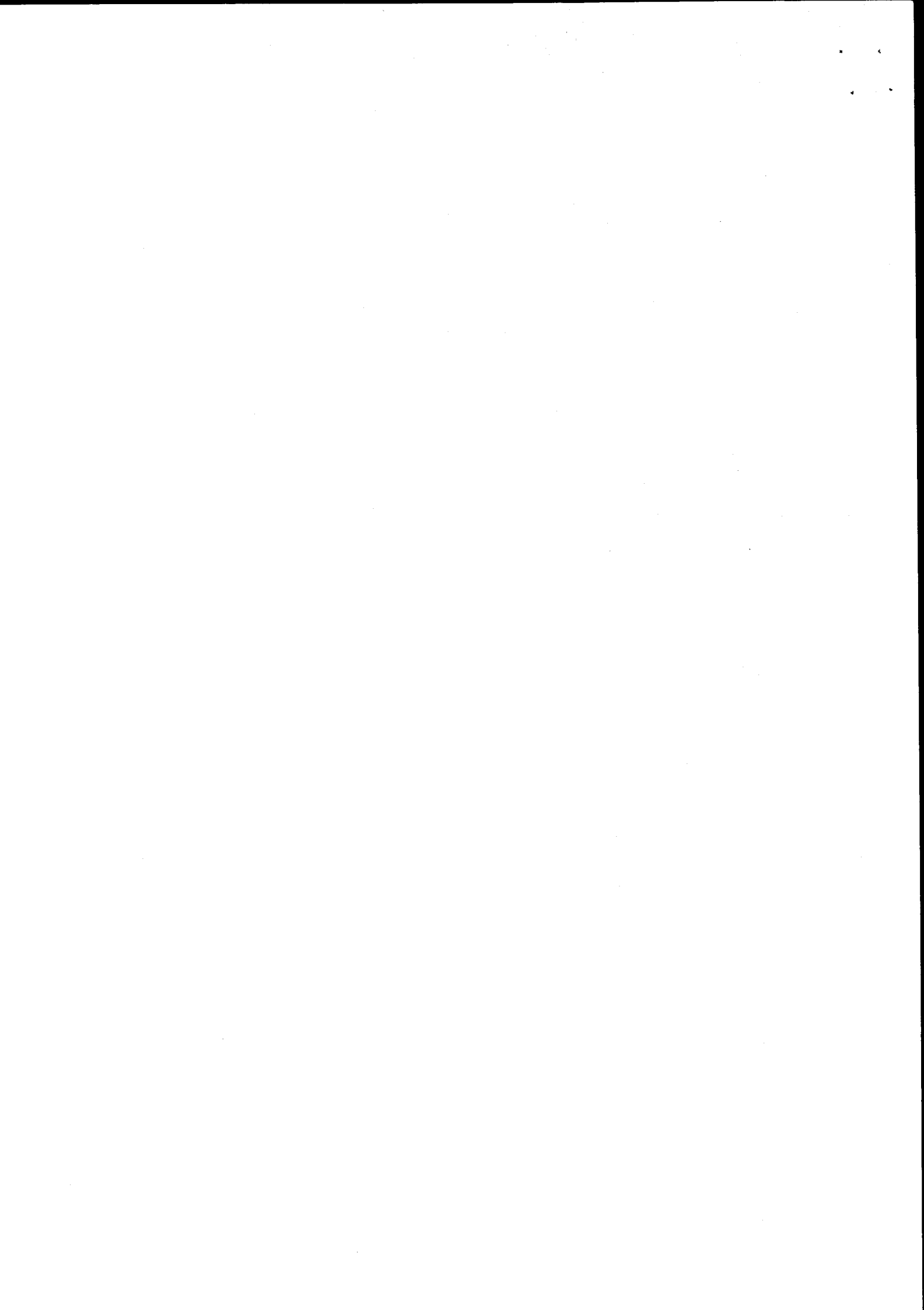
~~for i := 1 to c do~~ cor := 0;
 if a < b then c := a else if b < a then c := b else if b = a then c = a;

for i := 1 to c do begin
 if (a mod i = 0) ~~then cor := cor + 1;~~ (and (b mod i = 0)) then cor := cor + 1;
 end;

~~if cor = 1 then write('');~~
 if c = a then c1 = b else if c = b then c1 = a;

if cor = 1 then write(c - c1) else write('Не существует');
 end.

Т.Х.



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Димаксян

ИМЯ Марина

ОТЧЕСТВО Сейрановна

Дата рождения 04.01.1997_г

Класс: 11

Предмет информатика

Этап: заключительный

Работа выполнена на 3 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.



Язык Delphi 7.0.

```
1. var x: integer64; x: int64; i, q: integer;
      a: array [1..100000] of integer64; a: array [1..100000] of int64;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  read(x);
  a[1] := 1; a[2] := 2; a[3] := 3; q := 3;
  while (a[q] < x) do
  begin
    incl(q); a[q] := a[q-1] + a[q-2];
  end;
  if a[q] > x then dec(q);
  while (x > 0) do
  begin
    if (x - a[q]) >= 0 then begin x := x - a[q]; write(1) end else write(0);
    dec(q);
  end;
  for i := 1 to q do write(0);
end.
```

```
5. uses math;
var a: array [1..100000] of int64;
    m, n, x, q, i: integer;

procedure QSort (l, r: integer);
var i, j, x, y: integer;
begin
  i := l; j := r; x := a[(l+r) div 2];
  while (i <= j) do
  begin
    while (a[i] < x) do incl(i);
    while (a[j] > x) do dec(j);
    if (i <= j) do
    begin
      y := a[i]; a[i] := a[j]; a[j] := y;
      incl(i); dec(j);
    end;
  end;
  if (l < j) then QSort (l, j);
  if (i < r) then QSort (i, r);
end;
```



5 (продолжение).

```
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  read(m, n); x := m * n; q := 1;
  while q <= x do
    begin
      read(a[q]); inc(q);
    end;
  QSort(1, x);
  for i := 1 to n do
    begin
      for j := 1 to m do
        begin
          write(a[x], ' '); dec(x);
        end;
      writeln("");
    end;
  end.
```

```
2. var a: array [1..500000] of int64;
    b: array [30..1000000] of int64;
    N1, i, j, x, y, min: integer;
```

```
begin
  a[1] := 1; i := 2;
  while a[i-1] < N1 + 50 do
    begin
      a[i] := i * i;
      inc(i);
    end;
  for i := 1 to N1 + 50 do
    for j := 1 to N1 + 50 do
      for x := 1 to N1 + 50 do
        for y := 1 to N1 + 50 do
          inc(b[a[i] + a[j] + a[x] + a[y]]);
        end;
      end;
    end;
  end;
  min := 10000;
```



2 (продолжение)

```

for i := N1+1 to N1+50 do
  begin
    if b[i] > 1 and min > b[i] then min := b[i];
    inc(i);
  end;
for i := N1+1 to N1+50 do
  if b[i] = min then write (i, ' ');
end.

```

```

uses math, sysutils;
4. var m, n: array [0..1000] of integer;
    x: array [0..

```

```

procedure mas read (var b: array [0..1000] of integer);
var i;
begin
  i := 1;
  b[0] := 0;
  while (b[i-1] >= 0) and (b[i-1] <= 9) do
    begin
      read (b[i]);
      inc(b[0]);
      inc(i);
    end;
end;

```

```

procedure multimas (a, b, c: array [0..      ] of integer);
var i, j;
begin  FillChar(c, sizeof(c), 0);
  for i := b[0] downto 1 do
    for j := a[0] downto 1 do

```

Алгоритм предполагает использование массивов для хранения чисел, где $0 \leq i$ элемент является количеством цифр в числе. Заранее указывается массив X как $\{1, 1, 0, 0, \dots\}$ и раз первичными обратными массивов X и N . При этом в массиве M используя алгоритм вычитания длинных чисел. Смысл и цель описать.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант №

7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ДУРАКОВ

ИМЯ МАТВЕЙ

ОТЧЕСТВО ЕВГЕНЬЕВИЧ

Дата рождения 04.12.97

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 2 листах

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

Подпись участника олимпиады:



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



1) Для начала запишем в переменную N десятичное число

2) создадим переменную f и обнулим её (она будет содержать окончательное значение массива из 3х элементов $a[i]$ и вспомогательную переменную m число фибоначчи)

3) создаем цикл: `while N > 0 do begin` (выходим из цикла)
 `if N <= 3 then begin`
 `if N = 2 then f := f + 10 and break`
 `else if N = 1 then f := f + 1 and break`
 `end;`
 `else begin`
 `a[0] := 1;`
 `a[1] := 2;`
 `a[2] := a[0] + a[1];`
 `m := 2`
 `while a[2] <= N do begin`
 `a[0] := a[1];`
 `a[1] := a[2];`
 `a[2] := a[0] + a[1];`
 `m := m + 1;`
 `end;`
 `f := f + 10(m-1);`
 `N := N - a[1];`
 `end;`

в результате работы которого получим число f .

4) Выводим на экран число f : `writeln(f);`

5) 1) Создаем двумерный массив $a[M][N]$

2) запустим его так, как записано Монтей используя двойной (вложенный) цикл; например переменная i берает по строкам, а j по столбцам

3) присваивается переменной max значение $a[0][0]$;

4) с помощью двойного цикла ищем самый максимальный элемент
`for i := 0 to M do`

`for j := 0 to N do begin`

`if a[i][j] > max then begin`

`max := a[i][j]; k := i; l := j; end;`

5) меняем местами $a[0][0]$ и $(max) a[k][l]$ (значения этих ячеек)

6) создаем цикл от 1 до $N \times M$ в котором делаем двойной цикл по

поиску следующего максимального элемента, который может быть равен max и меняем его уже с $a[0][1]$ (где это $max := a[0][1]$ и делаем ищем элемент ~~большее~~ его и меняем их местами если такой есть, в результате работы этого алгоритма в конце концов мы получим отсортированный в порядке убывания ~~элемент~~ двумерный массив



№4 1) Если $N=0$, то $N^M=0$

2) Если $M=0$, то $N^M=1$ 3) Рассмотрим число $\overline{abcd} = N$

и $M = \overline{efgh}$, тогда

$$N^M = \overline{abcd}^{\overline{efgh}} = (a \cdot 1000 + b \cdot 100 + c \cdot 10 + d)^{\overline{efgh}} =$$

$$= a \cdot \overline{efgh} \cdot 10^{\overline{efgh}} + b \cdot \overline{efgh} \cdot 10^{\overline{efgh}} + c \cdot \overline{efgh} \cdot 10^{\overline{efgh}} + d \cdot \overline{efgh}$$

обратно, то $10^{\overline{efgh}}$ — это умножение на $10^{\overline{efgh}}$ и т.д.

т.к. $\overline{efgh} = M \Rightarrow$ при умножении числа X на $10^{\overline{efgh}}$ это будет

равносильно умножению на число X M раз: $X \underbrace{0000 \dots 00}_M$

при возведении же $a \cdot \overline{efgh} = a \cdot (e \cdot 1000 + f \cdot 100 + g \cdot 10 + h)$

$$= a \cdot e \cdot 1000 + a \cdot f \cdot 100 + a \cdot g \cdot 10 + a \cdot h, \text{ то получаем гораздо проще } \Rightarrow$$

можно рассмотреть M и N порознь \Rightarrow удобнее всего их хранить в соседних переменных.

№6 П.к. было выделено P сантиметров квадратов на каждого занимаем не более $2 \text{ м}^2 \Rightarrow$ все саженцы занимают не более $2 \cdot P \text{ м}^2 \Rightarrow$

т.к. участок был прямоугольный \Rightarrow его площадь была равна $K \text{ км} \cdot L \text{ км} = 1000 K \text{ м} \cdot 1000 L \text{ м} = 10^6 \cdot K \cdot L \text{ м}^2 \Rightarrow$

$2P \text{ м}^2 \leq 10^6 K \cdot L \text{ м}^2 \Rightarrow$ Я думаю буду идти по этому пути
 наименьшее из угла ^(сначала левая) выравню ориентиры относительно площади 2 м^2
 и если дерево есть, то занимаем в массиве $a[M][N]$
 (пробавить 1) к той же ячейке массива, в которой хранится число видов
 и если с нами не имеют друг друга ветви, но есть если встретится
 дерево с ^{другим} номером, то $a[M][N] := a[M][N] + 1$; с другой стороны
 я думаю создать двумерный массив размером $[K \cdot 1000][L \cdot 1000]$

в каждой ячейке от $[0][0]$ прибавляя по 2 метра $[1][0] \dots$
 я думаю буду заниматься 0, если нет (в этой ячейке) дерева нет и
 если оно есть, то я буду заниматься числом друзей у этого дерева в
 эту ячейку \Rightarrow в конце работы я получу по моему замечательный
 массив $a[]$ в каждой ячейке число оставшихся деревьев по видам и
 подробную карту $b[][]$ где какое дерево стоит и стоит ли вообще

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № _____

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ЖУНЕВ

ИМЯ НИКИТА

ОТЧЕСТВО АЛЕКСАНДРОВИЧ

Дата рождения 04.01.1998

Класс: 10

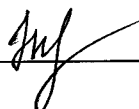
Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 2 листах

Дата выполнения работы: 03.03.2015
(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.



№3

1. Начало
2. Получаем входные данные: n, a .
3. Вводим переменные: $c = n$; $d = a$.
4. Если $n \neq d$, то 4.1, иначе 5
 - 4.1. Если $c > d$, то $c = c - d$, иначе $d = d - c$;
переходим в пункт 4.
5. Если $c = 1$, то $b = n - a$
6. Если $b \neq 0$, то вывод: b ; иначе вывод: не существует
7. КОНЕЦ

№4.

1. Начало
2. Получаем входные данные: n .
3. Вводим переменные: $i = 4$; $x = 0$.
4. Если $n > 2$, то переходим в 5, иначе переходим 4.1
 - 4.1. Если $(n = 1)$, то $x = 4$;
Если $(n = 2)$, то $x = 3$.
переходим в пункт 8.;
5. $n = n - 2$
6. Если $n > i$, то переходим в 6.1, иначе переходим в 7
 - 6.1. $n = n - i$;
 $i = i + 2$;
переходим в 6;
7. Если $(n = 1)$ или $(n = (i \text{ div } 2) + 1)$, то $x = 3$; иначе $x = 2$
8. Вывод x
9. Конеч.

№2

Преимущество двоично-десятичной системы счисления:

- 1) При переводе в десятичную систему счисления не теряется точность (для дробных чисел)
- 2) Упрощено умножение и деление на 10.
- 3) Упрощено округление.



№1.

1. Начало

2. Принимаем исходные данные: a .

3. Вводим переменные:

$$b = 0; i = 0; c = 0.$$

4. Если $a \neq 0$, то 4.1, иначе переходим 5

$$4.1. \quad b = b + ((a \bmod 10) \cdot 8^i);$$

$$i = i + 1;$$

переходим в 45. вывод: (значение в переменной b переводим в римскую систему счисления и выводим ответ)

6. Конец.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7101

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Керусеико

ИМЯ Алиса

ОТЧЕСТВО Алексеевна

Дата рождения 10.02.99

Класс: 10


Предмет информатика

Этап: II заключительный

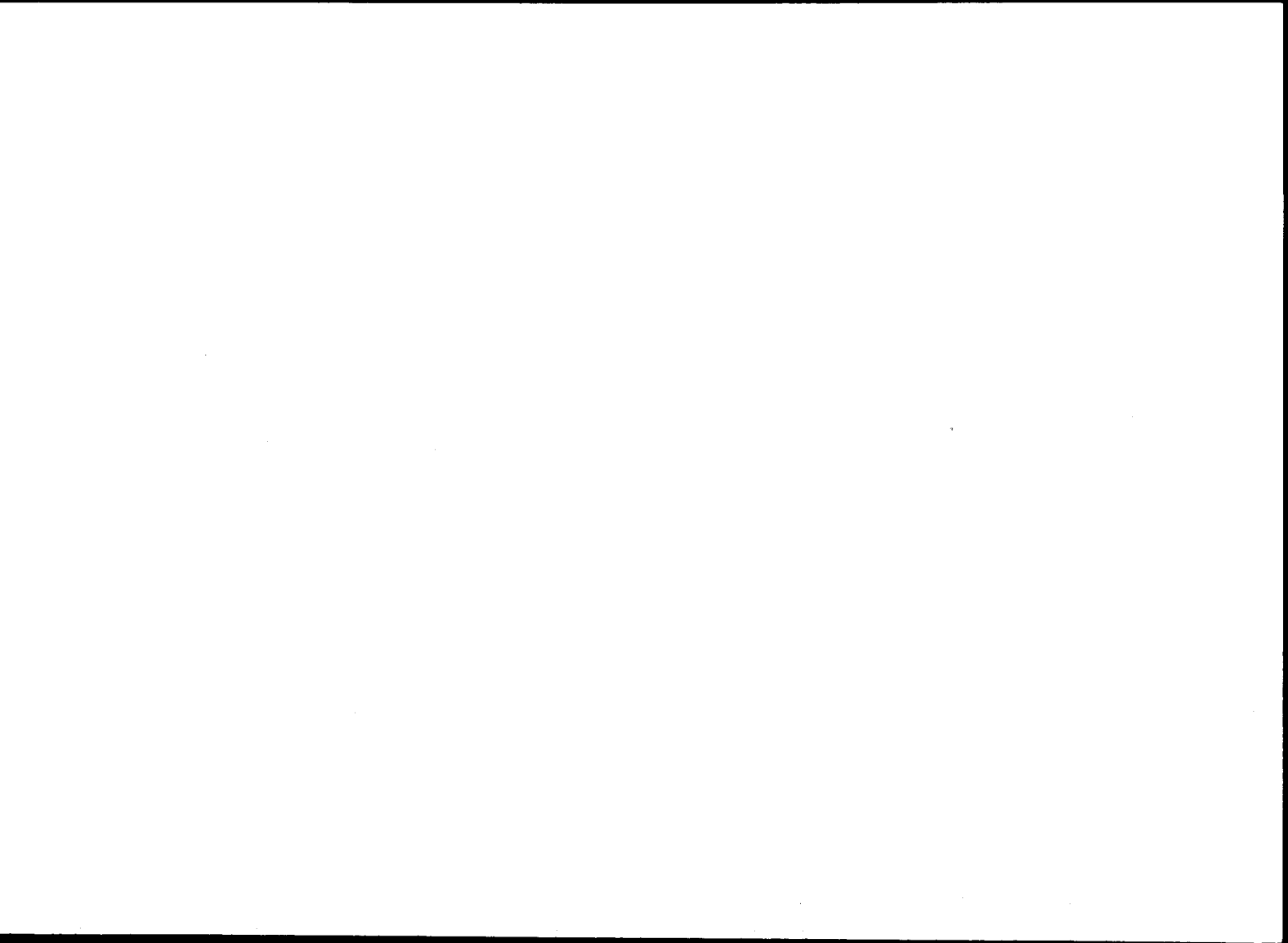
Работа выполнена на 1 листах

Дата выполнения работы: 20.05.15
(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
Начало
переменные: $n1, a1, c1, h1, m1,$
 $a2, h2, m2, a3, h3;$

$h1 := 1; m1 := 0$
переводим число в двоичную
систему счисления!
вводим число (n1)

пока $n1 <> 0$ выполняем:
 $h1 :=$ целая часть от деления
числа $n1$ на 10

$a1 :=$ остаток от деления
числа $n1$ на 10 ;
оператор выбора $a1$:

- 1: $c1 := 1$ 5: $c1 := 101$
- 2: $c1 := 10$ 6: $c1 := 110$
- 3: $c1 := 11$ 7: $c1 := 111$
- 4: $c1 := 100$

изменяем значения переменных
(т.к. невозможное число)
завершить оператор выбора.

$c1 := c1 \cdot h1$
 $h1 := h1 \cdot 1000$
 $m1 := m1 + c1$
завершить цикл пока.

переведем число в десяти-
ричную систему счисления: $h2 := 1$
пока $m1 <> 0$ выполняем:

$m1 :=$ целую часть от деления
числа $m1$ на 10
 $a2 :=$ остаток от деления
числа $m1$ на 10 ;

$m2 := m2 + a2 \cdot h2$;
 $h2 := h2 \cdot 2$;
завершить цикл пока.

переводим число в римскую
систему счисления:
 $h3 := 1$

пока $m2 <> 0$ выполняем:
 $m2 :=$ целая часть от деления
числа $m2$ на 10 ;

$a3 :=$ остаток от деления
числа $m2$ на 10 ;
 $a3 := a3 \cdot h3$

- оператор выбора $a3$:
- 1: выводим 'I'
 - 2: выводим 'II'
 - 3: выводим 'III'
 - 4: выводим 'IV'
 - 5: ... 'V'
 - 6: ... 'VI'
 - 7: ... 'VII'
 - 8: ... 'VIII'
 - 9: ... 'IX'
 - 10: ... 'X'
- 20: ... 'XX'
 - 30: ... 'XXX'
 - 40: ... 'XL'
 - 50: ... 'L'
 - 60: ... 'LX'
 - 70: ... 'LXX'

- 80: ... 'LXXX'
- 90: ... 'XC'
- 100: ... 'C'
- 200: ... 'CC'
- 300: ... 'CCC'
- 400: ... 'CD'
- 500: ... 'D'
- 600: ... 'DC'
- 700: ... 'DCC'
- 800: ... 'DCCC'

№5
таблица имеет вид:

значение параметра	результат
a	10
b	5
...	...

f - номер
столбца
c - номер
строки

для c от 2 до количества
переменных выполняем
цикл:
если число, расположенное
в координатах (f; c), меньше
числа, расположенного в коор-
динатах (f; c+1), то меняем
результаты этих переменных
и меняем значение
параметра в координатах
(f+1; c) и (f-1; c+1)
конец цикла.

выполняем цикл до тех
пор, пока элемент в ко-
ординате (2; 2) не станет
минимальным.

- 900: ... 'CM'
- 1000: ... 'M'
- 0: ... '' (нигило)
- и т.д.
- завершить оператор
выбора
- $h3 := h3 \cdot 10$;
- завершить цикл пока
- КОНЕЦ

см. на обороте

№6

Назано
переменные: два массива a и b

i - целое натуральное число

S_1 и S_2 - натуральные числа

1) заполним массив для первого
игрока (Петя):

для $i :=$ от 1 до количества партий
выполняем:

$a[i] :=$ количество очков, выбран-
ных Петей в $[i]$ -ой партии.

2) сложим все элементы массива a :

для $i :=$ от 1 до количества партий
выполняем:

$S_1 := S_1 + a[i]$

3) заполним массив для второго
игрока (Данила):

для $i :=$ от 1 до количества партий
выполняем:

$b[i] :=$ количество очков, выбран-
ных Данилой в $[i]$ -ой партии.

4) сложим все элементы массива b :

для $i :=$ от 1 до количества партий
выполняем:

$S_2 := S_2 + b[i]$

5) если $S_1 < S_2$ выводим 'Петр'
иначе выводим 'Данил'

№2

Двоично - десятичный код
целесообразно использовать
там, где есть необходимость
этого использования про-
цедуры ввода - вывода. В таких
устройствах, как: электронные
часы, калькулятор и т.д.

В этих устройствах не всегда
целесообразно предусматри-
вать универсальный микро-
код, по причине небольшого
объема сохраняемой инфор-
мации.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ КОНОВАЛОВ

ИМЯ СЕМЕН

ОТЧЕСТВО АЛЕКСАНДРОВИЧ

Дата рождения 18.03.1997

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.



Задача 5.

Программа на C++

```
#include <iostream>
#include <cmath>
#include <vector>
#include <algorithm>
#include <stdio.h>
using namespace std;
int main() {
    long long n, m, x; // объявим переменные типа long long
    vector <long long> vp;
    cin >> m >> n; // считаем размер строки
    for (int j=0; j < m*n; ++j) // в цикле пройдем до конца
    { // считаем числа строки
        cin >> x; // занесем их в вектор
        vp.push_back(x);
    }
    sort(vp.begin(), vp.end()); // посортируем их в порядке
    for (int j=0; j < m*n; ++j) // убывающе
    {
        cout << vp[j] << " "; // выведем числа в убывающем
        if (!(j % m)) cout << endl; // порядке через
        // строку каждое
        // число
    }
    return 0;
}
```

Задача 3.

Программа на C++

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cmath>
#include <sstream>
using namespace std;
long long func(int x)
{
    if ((x > 3) && ((x % 6) && 3) != 1) return 0;
    for (long long q=4; m=5; pow(m, 2) <= x; m=m+q^6)
```



```
if (x % m == 0) return 0; return 1; }  
// эта функция проверит данное нам число простое или  
// не простое  
int main() {  
    long long p;  
    cin >> p;  
    if (func(p)) printf("Вероятно простое");  
    else printf("Составное");  
    // проверка числа и вывод ответа  
    return 0;  
}
```

Задача 1.

Программа на C++

```
#include <iostream> // подключаем библиотеки  
#include <vector>  
#include <string>  
#include <algorithm>  
#include <cmath>  
using namespace std;  
int main() {  
    long long a, f = 0;  
    vector<long long> vp;  
    cin >> a;  
    vp.push_back(1);  
    vp.push_back(1);  
    while (f <= a)  
    {  
        f = vp[vp.size()-2] + vp.back();  
        if (a >= f) vp.push_back(f);  
    } // перебор чисел Фибоначчи меньше заданного числа  
    string str = "";  
    reverse(vp.begin(), vp.end());  
    for (int j = 0; j < vp.size(); ++j) // определяем какие из чисел  
    { if (a - vp[j] >= 0) str += '1'; // Фибоначчи используются  
      else str += '0'; // последовательности добавляем  
      // из нашего исходного числа.  
      if (str[str.size()-1] == '1') a -= vp[j];  
    }  
    cout << str; // выводим ответ
```



```
return 0;
```

```
}
```

Задача 4

Программа на языке C++

```
#include <iostream> // подключаем библиотеки
```

```
#include <algorithm>
```

```
#include <cmath>
```

```
#include <cstdio>
```

```
using namespace std;
```

```
int main () {
```

```
long long a, b, c, mas[10005];
```

```
cin >> a >> b;
```

```
c = 1;
```

```
mas[1] = a;
```

```
for (int i = 2; i < b; ++i)
```

```
{ for (int j = 1; j < c; ++j)
```

```
{ mas[j] = mas[j] * a;
```

```
for (int j = 1; j < c; ++j)
```

```
{ if (mas[j] > 9) { mas[j+1] = mas[j+1] +  
+(mas[j] / 10);
```

```
mas[j] = mas[j] % 10;
```

```
if (c < j+1) c = j+1;
```

```
}
```

```
}
```

```
for (int i = c; i > 1; --i)
```

```
{ cout << mas[i];
```

```
}
```

```
return 0;
```

```
}
```



Задача 6

Программа на языке C++

```
#include <iostream>
#define ll long long
using namespace std;
int main() {
    ll a, b, c, d m, n, k, l, p, x;
    cin >> m >> n >> p >> k >> l;
    ll mas[m]; // задаем массив счетчиков еще по количеству
                // друзей
    for (int i = m; i <= n; ++i) // обнуляем массив
        { mas[i] = 0;
        }
    for (int i = 0; i < p; ++i) // считываем количество еще с цен
        { cin >> x; // дружим и массив счетчиков
          mas[x] ++; // увеличиваем.
        }
    ll b = m;
    for (int i = 1; i <= k; i = i + 2) // заносим в таблицу количество
        { for (int j = 1; j <= l; j = j + 2) // еще с определенными друзьями
          { cout << mas[b];
            b ++;
          }
        }
    cout << endl;
    return 0;
}
```

Задача 2

Программа на языке C++

```
#include <iostream>
using namespace std;
int main() {
    long long a, n;
    cin >> n;
    for (int a = 1; a <= 2; ++a)
        for (int b = 1; b <= 3; ++b)
```



```
for (int c=1; c<=5; ++c)
  for (int d=1; c<=10; ++ ++d)
    { q = a * a + b * b + c * c + d * d;
      if (q>n)
        if (q<=n+50)
          cout << q << " ";
        else return 0;
    }
return 0;
}
```

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7111

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ КОРШУНОВ

ИМЯ ИГОРЬ

ОТЧЕСТВО ЕВГЕНЬЕВИЧ

Дата рождения 19.04.1997

Класс: 11

Предмет информатика

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

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

Подпись участника олимпиады: *Коршунов*

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

п.п. в условии сказано, что на листке с отформатированной таблицей $M \times N$ клеток инициализировано значение 3, а известно, что на строке $M \times N$ клеток не превосходит значение двух и более строк таблицы. В этом случае мы можем перебрать размерами строк и столбцов и считать все эти варианты $n \gg x$, т.е. пустой клетки (в начале условия в клетке, а используемая всего 3, значением 2) будем пропускать.

Код программы на C++:

```
#include <fstream>
```

```
using namespace std;
```

```
void main()
```

```
{
    ifstream in; in.open("input.txt");
    ofstream out; out.open("output.txt");
    int number, rez1, rez2, rez01, rez02, M, N;
    in >> M >> N >> endl;
    rez01 = 0; // Если рассмотреть, что количество ячеек
    rez02 = 0; // занимаем ровно 1 клетку, то это
    while (in >> number) // выведем сколько раз в строке из M символов
    { // поделится на 3 ячейки и обработается
        in >> rez1; // т.е. * будет код который будет после
        rez01 += rez1; // обработанной переменной:
        in >> rez2; // int i, j, k;
        rez02 += rez2; // in >> M >> N >> endl;
        in >> endl; // rez01 = 0;
        // rez02 = 0;
    } // k = M / 3;
    if (rez01 > rez02) // for (i = 0; i < N; i++)
    { // {
        out << "ПЕТР"; // for (j = 0; j < k; j++)
    } // {
    else // {
    { // in >> number >> rez1 >> rez2;
        out << "Данил"; // rez01 += rez1;
        // rez02 += rez2;
    } // }
    // in >> endl;
    // }
    // if (rez01 > rez02)
    // {
    //     out << "ПЕТР";
    // }
    // else
    // {
    //     out << "Данил";
    // }
    // }
    // }
}
```

P.S. п.п. в условии не было сказано как поступить в случае равенства баллов, написанный мной код будет принят. Занять победу Данилу.



N2

Использование двоично-десятичной системы числения является целесообразным в том случае, когда при работе с данными, нам удобнее использовать десятичную кодировку каждого, но есть необходимость работы с числом как с десятичной разрядной шкалой. Например перед нами стоит задача отсортировать разряды десятичного числа, на месте которых стоят цифры и вывести на экран двоичный код каждой цифры которая осталась в числе.

N4

сначала определили разрядность квадрата, который будем обрабатывать так как в задаче N2 мы уже, затем определили количество цифр которые необходимо вывести из оставшегося количества клеток за пределами квадрата.

код C++:

```
#include <fstream>
using namespace std;
void main()
{
    ifstream in; in.open("input.txt");
    ofstream out; out.open("output.txt");
    int k, N, x, z;
    in >> N;
    if (N == 1)
    {
        out << "4";
    }
    else
    {
        k = 0;
        while (k * k < N)
        {
            k++;
        }
        if (k * k == N)
        {
            x = 0;
            x = k * (k + 1) * 2;
            out << x;
        }
        else
        {
            k--;
            x = k * (k + 1) * 2;
```

```

x = x + 3;
z = N - k * k - 1;
if (z < k)
{
    x = z * 2;
}
else
{
    x = z k * 2 + 1;
    z = z - k;
}
if (z > 0)
{
    x += z * 2;
}
out << x
}
}
}
```



№5

существует несколько структур для хранения данных, наиболее простая структура - массив. ~~В. Чудов~~
 т.к. наша таблица имеет как текстовые поля (значения параметров), так и числовые, мы будем использовать для хранения два массива, один двумерный (числовой), в котором будут храниться номера ответов результатов. Вторым (строковым) будем содержать значения полей "значения параметров". Т.к. работа с массивами проще в языке Pascal (имеется стандартный тип данных string) для экономии времени код будем написать именно на нем. Для сортировки будем использовать сортировку quick sort, однако напишем сортировку на более простом и быстрее методом пузырька (время работы намного меньше кода).
 P.S. На протяжении "значения параметров" отбрасываем пробелы. В языке Pascal мы не можем задать максимальное значение (можем дать число) или обеспечить, что в базе данных не будет всевозможные символы записаны как пробелы (когда зная длину меньше максимальной длины) мы можем считать все строки и произвести сортировку, в противном случае строки будут перемешаны и неизвестно какой сортировки.

```

var mas1: array [1..100] of string [10];
    mas2: array [1..100, 1..2] of integer;
    integer: i, j, N, K;
    string u;
begin
  assign(input, 'input.txt'); reset(input);
  assign(output, 'output.txt'); rewrite(output);
  readln(N);
  while (not eof(input)) do
  begin
    N := N + 1;
    readln(mas2[N, 1]);
    readln(mas1[N]);
    readln(mas2[N, 2]);
  end;
  for i := 1 to N do
  begin
    for j := i + 1 to N do
    begin
      if mas2[i, 2] > mas2[j, 2] then
      begin
        mas2[i, 1] := mas2[j, 1];
        mas2[i, 1] := mas2[j, 1];
        mas2[j, 1] := mas2[i, 1];
        mas2[j, 2] := mas2[i, 2];
        mas2[i, 2] := mas2[j, 2];
        mas1[i] := mas1[j];
        mas1[j] := mas1[i];
      end;
    end;
  end;
end;

```

```

for i := 1 to N do
begin
  write(mas2[i, 1], ' ');
  write(mas1[i], ' ');
  writeln(mas2[i, 2]);
end;
end.

```

P.S. На протяжении "значения параметров" отбрасываем пробелы. В языке Pascal мы не можем задать максимальное значение (можем дать число) или обеспечить, что в базе данных не будет всевозможные символы записаны как пробелы (когда зная длину меньше максимальной длины) мы можем считать все строки и произвести сортировку, в противном случае строки будут перемешаны и неизвестно какой сортировки.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7111

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ КУДИНОВ

ИМЯ ДАНИИЛ

ОТЧЕСТВО Сергеевич

Дата рождения 14.05.1998

Класс: 11


Предмет ИНФОРМАТИКА

Этап: II ЗАКЛЮЧИТЕЛЬНЫЙ

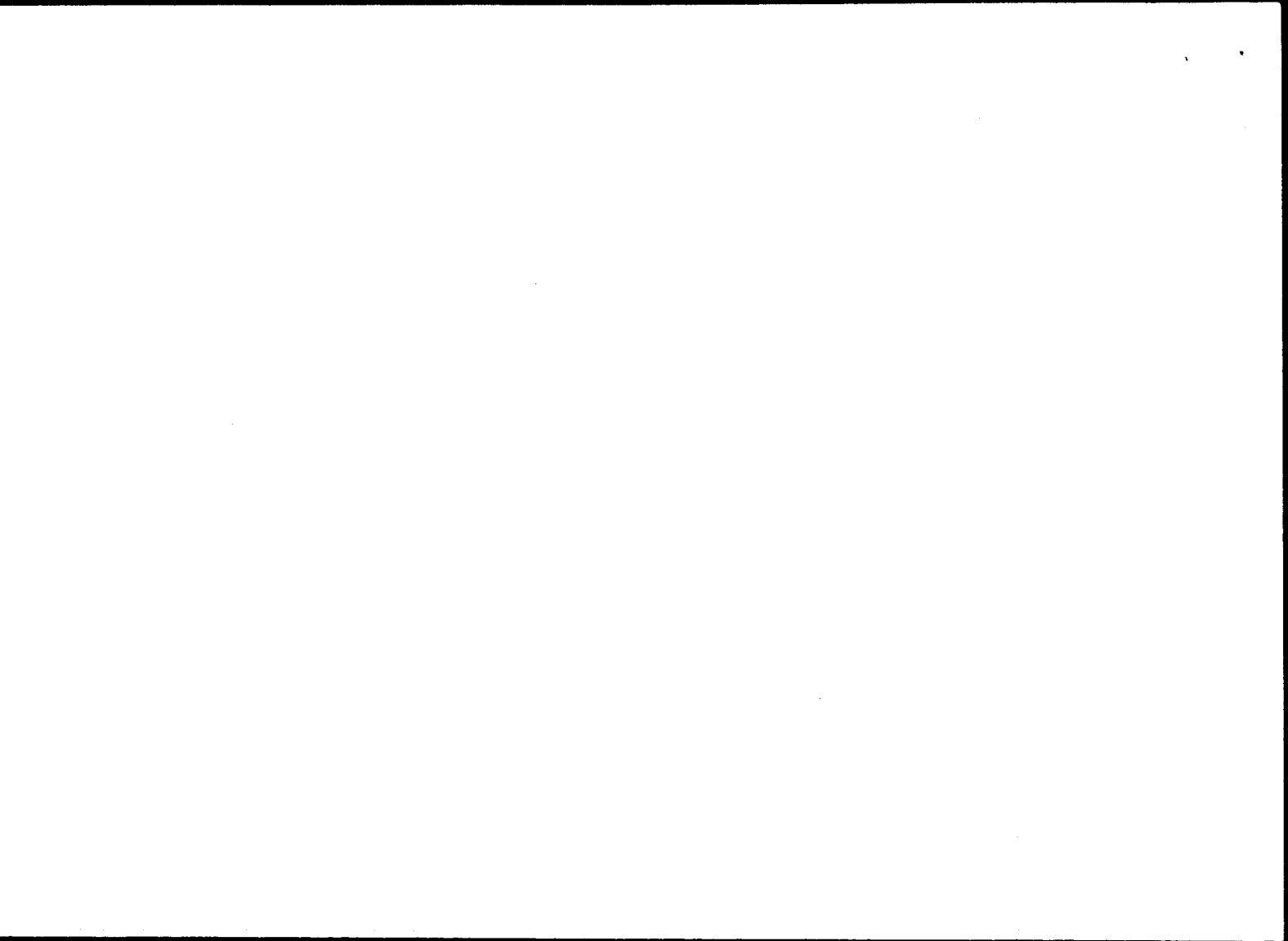
Работа выполнена на _____ листах

Дата выполнения работы: 20.03.2015
(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 * В алгоритме в фигурных скобках $\{ \}$ даны мои комментарии, т.к. обычные скобки могут использоваться в математических целях.

рассмотрим алгоритм на примере двузначного числа:
 Принцип алгоритма — расписать цифру каждого разряда как набор римских символов (естественно, учитывая сам разряд — от него зависит, какие символы будут использоваться).

Таким образом достаточно составить алгоритм для двузначного числа — для чисел с большим кол-вом разрядов алгоритм изменится «шаблонно».

Алгоритм для 2-значного числа:

1) Берем целую часть от деления числа на 10. x (обозначим x)

2) Если $x \geq 5$ — пункт 2.1. Иначе — пункт 2.2.

2.1) Если $x \geq 5$ — пункт 2.1.1. Иначе — пункт 2.1.2.

2.1.1) пишем сперва \overline{X} $(10-x)$ раз, потом M {имеется в виду, что M в римском обозначении — 100. Я могу ошибаться}.

2.1.2) пишем сперва L { $L=50$ }, после — \overline{X} $(x-5)$ раз.

2.2) $5-x \geq 2$? Если да — пункт 2.2.1, Нет — 2.2.2.

2.2.1) пишем \overline{X} x раз.

2.2.2) пишем сперва \overline{X} $(5-x)$ раз, потом L . {на сколько я знаю, в римском обозначении допускается подряд не более 2х одинаковых символов. Исключение — \overline{III} . В алгоритме используются именно эти условия. Если же допускается 3 символа, двойка в условии пункта 2.2 меняется на тройку}.

3) Обнимаем наш x и обозначаем им остаток от деления на 10.

4) Если $x \geq 3$ — пункт 4.1. Иначе — пишем I x раз.





4.1) Если $x75$ - пункт 4.1.1. Иначе - пишем „IV“.

4.1.1) Если $x-572$, пишем сперва „I“ $(10-x)$ раз, потом „X“. Иначе - пишем „V“, потом „I“ $(5-x)$ раз.

5) Вывод числа. Конеч.

Как видно - блок 2 - преобразование 2000 разряда (10), а блок 4 - преобразование 1000 разряда (1). Блоки для сотен, тысяч, десятков тысяч будут идентичны блоку 2, в котором лишь изменятся символы. Таким образом, универсальный алгоритм можно начать с определения старшего разряда числа (чтобы понять, с какого разряда начинать работу).

№2

Опытным путем было выяснено, что количество битов по обычной двоичной записи числа никогда не будет ~~большее~~ ^{большее} количества битов для записи числа в системе 2-10. Т.е. для экономии места этот способ не подходит. Возможно, применение этой системы будет целесообразно, если в числах присутствует много повторяющихся цифр - путем их группировки (некто вроде архивации) их вес можно уменьшить. Например, число 3000000 будет закодировано как 0011 и 0000 ^{браз.} В конце концов можно создать „базу“ с цифрами от 0 до 9 и, при записи большого кол-ва больших цифр, ссылаться на элементы этой базы (массива). Так будет экономлено большое кол-во памяти.



н3

1) Для начала, внесем, взаимно простые ли модуль и число. Это можно сделать разбиванием на множители (делим и модуль и число на целые числа, начиная с 2, на которые они делятся нацело, и ищем, есть ли среди них одинаковые (сравним ~~данные~~ множители модуля с множителями числа)).

2) Если они взаимнопростые — выполняем далее. Иначе — конец алгоритма и вывод «Не существует!».

3) Теперь выполним своеобразную процедуру: умножаем модуль сначала на 2, прибавляем 1, делим на число, данное нам вначале вместе с модулем. Если число целое — это и есть наш результат работы — второе число. Если число выходит не целое — умножаем наш модуль теперь на 3, снова прибавляем 1 и т.д. Т.е. увеличиваем множитель для модуля, пока, в результате операции не получим целое число.

4) ~~Конец~~. Вывод результата. Конец.

На примере чисел 32 и 9 (32 — модуль):

$$32 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \quad \text{— одинаковых множителей нет.}$$

$$9 = 3 \cdot 3$$

~~32~~ $32 \cdot 2 + 1 = 65$ — на 9 не делится.

$32 \cdot 3 + 1 = 97$ — на 9 не делится.

$32 \cdot 4 + 1 = 129$ — на 9 не делится

$32 \cdot 5 + 1 = \dots$

\dots
 $32 \cdot 7 + 1 = 225$ — делится на 9 $= 25$

25 — мультипликативно обратное числу 9 число. По модулю 32

л4

Удобнее всего воспользоваться циклом:

~~Пока $N > 0$ выполнять:~~

Для начала, посчитаем кол-во рядов (k) где первых двух квадратов (они не попадают под закономерность):

$$k := 0$$

$$N - 1 \geq 0?$$

Если да $k := k + 1$, иначе — рядом квадраты выкладывать не собираемся.

$$N := N - 1$$

$$N - 1 \geq 0?$$

Если да $k := k + 3$, иначе — ответ — 4 ряда и конец.

$N := N - 1$; $i := 2$; $n := 2$; { n будет считать шаги в цикле (отмерять квадраты), i будет использоваться для закономерности }
нахождение k

Далее идет сам цикл:

Пока $N > 0$, выполнять:

Начало Ц.

$$N := N - n \cdot 2;$$

$$\text{если } N \geq 0, \text{ тогда } k := k + 6 + i \cdot 2;$$

иначе,

$$\text{если } 2n - N < n, \text{ то } k := k + 3 + (N - 1) \cdot 2;$$

$$\text{иначе, } k := k + 6 + (N - 2) \cdot 2$$

$$i := i + 2$$

$$n := n + 1$$

Конец Ц.

Вывод k

Конец.

л5

- 1) сравниваем 1-ю колонку результатов со 2-й (далее элементы будут обозначаться просто цифрами).
172? да - пункт 2; нет - пункт 3.
- 2) если да, меняем их местами и повторяем п.1.
- 3) если нет, сравниваем 2 с 3. 273? да - п.4, нет - п.5
- 4) если да - сравниваем 3 с 4 ~~и т.д.~~
- 5) если нет - меняем 2 и 3 местами и начинаем сначала с 1 пункта.

Видно, что алгоритм рекурсивный. Это не очень удобно. Лучше заменить его циклом, в котором при определенных условиях будут меняться значения сравниваемых чисел. Например:

 $n := 1$ $m := 2$ {нач. значения сравниваемых показателей}

НЦ

если $n > m$, меняем их местами и $n := 1$, $m := 2$ иначе, $n := n + 1$, $m := m + 1$

КЦ

Также необходимо в конце цикла проверить, готова ли работа, т.е. сравнить по очереди все элементы ~~в~~ колонки результатов. Если где-то встретится элемент, больший чем следующий — цикл продолжает работу. Иначе — конец. Упорядочение завершено
прим.: когда мы меняем местами результаты в колонке, мы меняем всю строку (попытка и, значение параметра⁵).

лб

Необходимо сложить ^{все} элементы столбца "рез-Т1", потом сложить элементы столбца "рез-Т2". Сравнить два полученных значения и вывести имя того, ~~чей~~ чья сумма оказалась больше.

Например, в языке Pascal это будет выглядеть примерно так (таблица представлена двумерным массивом):

```
Program res;  
var a: array[1..3, 1..n] of integer;
```

```
    n, i, s1, s2: integer;
```

```
Begin n := 50; s2 := s1;
```

```
  readln(n);    { ввод кол-ва партий }
```

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

```
    begin  
    readln(a[2, i]);  
    readln(a[3, i]);  
  end;
```

^{и рез-Т2}
{ ввод рез-Т1. Ввод номеров партии пропускается — он здесь не необходим }

```
  for i := 1 to n do  
    begin  
      s1 := s1 + a[2, i];  
      s2 := s2 + a[3, i];  
    end;
```

```
  if s1 > s2 then
```

```
    writeln('Петя'); { не было указано в задании, чей столбик  
    результатов кому принадлежит. Допустим, что рез-Т1 — результаты Пети }
```

```
  else writeln('Данил');
```

```
end.
```

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № _____

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ НОВИКОВ

ИМЯ СЕРГЕЙ

ОТЧЕСТВО ИГОРЕВИЧ

Дата рождения 15.02.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 8 листах

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

Подпись участника олимпиады: Глеф

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. Язык программирования C++:

подключаем библиотеки:
#include <iostream> // ввод-вывод
#include <cmath> // мат-ие операции
#include <cstdlib> // random
#include <ctime>

using namespace std;

функция быстрого возведения в степень:

```
int pow (int a, int n) {  
    int res = 1;  
    for (; n; n >>= 1) {  
        if (n & 1) res *= a;  
        a *= a;  
    }  
    return res;  
}
```

Главная функция программы:

```
int main() {  
    int p, a, d, s, t, b, l, c = 0;  
    cin >> p; // ввод данных;  
    b = p - 1;  
    for (int i = 1; (2 << i) < b; i++)  
        if ((b % (2 << i) == 0) && ((b / (2 << i)) & 1))  
            d = b / (2 << i), s = i;  
    // перебираем s и возводим d;  
    l = t = log(p); // возводим t (глобальной log)
```



```
 srand(time(0)) // инициализируем random;
 while (L--){
   a = rand()%p; // выборка t случайного
                                     a ∈ {1, 2, ..., p-1}
   if (pow(a, d)%p == 1) c++; // проверка первого
                               условия для a;
   for (int i=0; i<S; i++){
     #f = pow(a, (2<<i)*d);
     if (f < 0 && abs(#f)%p == 1){
       c++; break; // проверка
                   второго условия;
     }
   }
 }
 }
```

Если p - простое, то выполнится одно из условий и c увеличится на единицу при любом a , но если ровно $t(1)$ раз, это и проверим при входе:

```
 cout << (c == t ? "Вероятно простое" : "Составное");
 return 0; // главная функция возвращает 0;
 }
```

5. для решения в нашей клемме:

```
 #include <iostream> // flog-folog
 #include <algorithm> // STL-алгоритмы
```

```
 using namespace std;
```

```
 int main(){
```

```
   int n, m, a[100500]; // ограничения не даны, так
                       что n*m = 100500;
```



```
cin >> m >> n; // пусть массив будет m строк
                // по n элементов;
for (int i=0; i < m*n; i++)
    cin >> a[i]; // считываем m*n
                // чисел в одномерный массив;
sort (a, a + m*n, greater <int>()); // сортируем
                // массив;
for (int i=0; i < m*n; i++){
    if (i != i%n == 0) cout << endl;
                // через каждые n элементов
                // делаем перевод строки;
    cout << a[i] << ' '; // выводим числа через
                // пробел;
}
return 0;
}
```

```
1. #include <iostream> // влог - вывод
    #include <vector> # // подключение вектора
    #include <string> // подключение строки
    #include <algorithm> // STL
    #define p-b(push-back) // замена стандартной
                            // записи на короткую.
using namespace std;

int main(){
    vector <int> f;
    int m, a=1, i=2;
```




```
string s = "" *; // создаём пустую строку
cin >> m; // вход переводимого числа

f.p-b(1), f.p-b(1); // добавление первых двух чисел
Фибоначчи;
```

```
for (; m <= 0; i++) {
    a = f[i-2] + f[i-1];
```

```
for (; a <= m; i++) {
    a = f[i-1] + f[i-2];
    f.p-b(a);
```

```
} // заполнение вектора f
числами Фибоначчи.
```

```
for (int j = f.size()-1; j >= 0; j--) {
    s += (m - f[j] >= 0 ? '1' : '0');
    m -= (m - f[j] >= 0 ? f[j] : 0);
```

```
} // возвращаем
ответ;
```

```
if (s[0] == '48') {
    cout << 0;
    return 0;
```

```
}
cout << s;
return 0;
```

```
}
```

```
6. #include <iostream>
#include <memory.h>
using namespace std;
```



```
int ms[100000][100000]; // k ≤ 100; l ≤ 100;

int main() {

    int m, n, k, l, p, a; ms[500][500];;
    cin >> m >> n >> p >> k >> l; // вход данных;
    int mm[A+1]; // создаём массив из A+1 элементов;
    memset(mm, 0, sizeof(mm)); // обнуляем массив,
                                // который будет счётчиком;
    for (int i=0; i<p; i++){
        cin >> a;
        mm[a]++;
    }
    int t=m; // считываем вход ещё
              // и считаем их кол-во;

    Вставляем поуровневый участок с явным,
    которые есть сейчас:
    memset(ms, 0, sizeof(ms));
    for (int i=1; i<=k; i+=2)
        for (int j=1; j<=l; j+=2){
            if (mm[t]) {
                ms[i][j] = t;
                mm[t]--;
            }
            if (mm[t] == 1) {
                ms[i][j] = t;
                mm[t]--;
                t++;
            }
        }
    if (!mm[t]) t++;
}
```



```
return 0;
```

```
}
```

```
4. #include <iostream>
#include <
#define ull unsigned long long
using namespace std;
```

```
ull pow(ull a, ull n){
    ull res = 1;
    for (; n; n >>= 1){
        if (n & 1) res *= a;
        a *= a;
    }
    return res;
}
```

```
}
```

```
int main(){
```

```
long long x, y, z, m [10005];
```

```
cin >> x >> y;
```

```
ull a, ull b;
```

```
cout << pow(a, b);
```

// решение работает только для

$a, b \leq 10^{20}$

(20 цифр) примерно.



```
9. #include <iostream>
#include <string>
#include <algorithm>
#include <vector>
using namespace std;
bool cmp(string a, string
int main(){
int n, m;
cin >> m >> n;
using namespace std;
bool f(string a, string b){
    int n = a.size();
    for (int i = 0; i < n; i++)
        if (a[i] > b[i]) return true; else
return true;
        if (a[i] < b[i]) return false;
    return true;
}
bool cmp(string a, string b){
    return ((a.size() > b.size()) || (a.size == b.size
        && f(a, b))) ? 1 : 0;
}
int main(){
    int c = 0;
    string s;
    vector<string> v;
    while (cin >> s)
        v.push_back(s);
```



```
sort(v.begin(), v.end(), cmp0);  
for (int i=0; i<v.size(); i++)  
    c += v[i].size()+1;  
    cout << v[i] << ' ' ;  
    if (c + v[i+1].size() > n) {  
        cout << endl;  
        c = 0;  
    }  
}  
}  
return 0;  
}
```

Олимпиада школьников «Надежда энергетики»

№ группы

шифр

← Не заполнять
Заполняется
ответственным
работником

Вариант № 7111

ФАМИЛИЯ ОПАЛЕВ

ИМЯ Антон

ОТЧЕСТВО Викторович

Дата рождения 12.08.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: 2

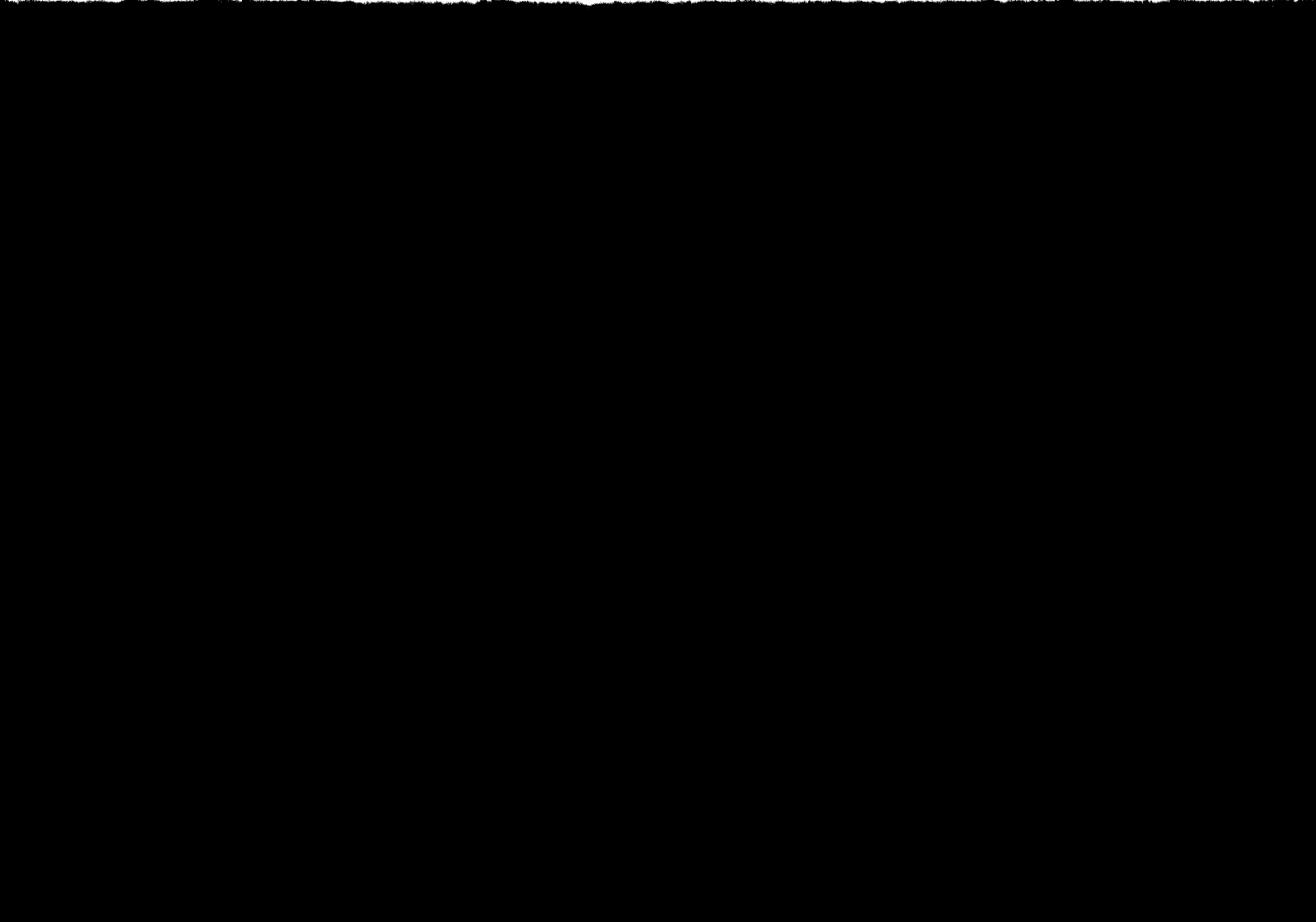
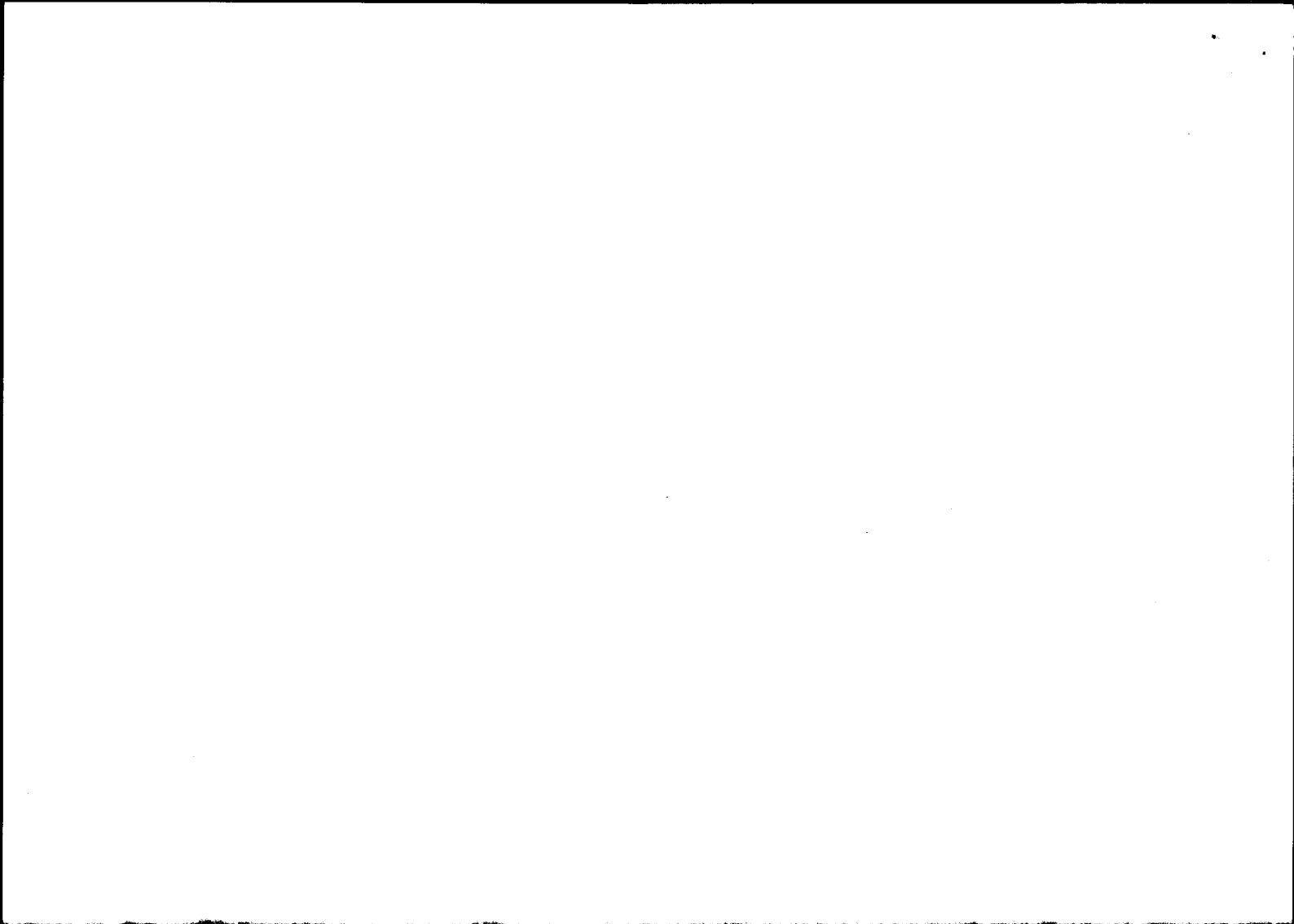
Работа выполнена на 2 листах

Дата выполнения работы: 4.03.15.
(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.





```
N6 Program: igra;
Var
M: integer;
N: integer;
Petr: integer;
Danil: integer;
c: integer;
a: array[1..M], [1..N] of integer;
```

Begin.

```
readln(M);
readln(N);
```

```
Petr:=0;
```

```
Danil:=0;
```

```
c:=1;
```

Repeat

```
Petr:=Petr+a[c,2];
```

```
Danil:=Danil+a[c,3];
```

```
c:=c+1.
```

```
Until a[c,1]=0;
```

```
if Petr > Danil then writeln('Петр');
else writeln('Данил');
```

```
if Petr = Danil then writeln('Ничья');
```

End.

```
N7. Program: chisto;
```

Var

```
a: array[1..13] of string;
```

```
b: array[1..13] of integer;
```

```
i, n, k: integer;
```

```
c, chisto: string;
```

Begin

```
c := 'MCMDCXCXLXIXVIXI';
```

```
for i: 1 to 13 do;
```

Begin

```
if i mod 2 <> 0 then k:=1 else k:=2;
```

```
a[i]:=copy(c, 1, k);
```

```
case i of
```

Вводим размер страницы.

Петру и Данилу присваиваем по 0 очков.


```

1 b[i] := 1000;
2 b[i] := 900;
3 b[i] := 500;
4 b[i] := 400;
5 b[i] := 100;
6 b[i] := 90;
7 b[i] := 50;
8 b[i] := 40;
9 b[i] := 10;
10 b[i] := 9;
11 b[i] := 5;
12 b[i] := 4;
13 b[i] := 1;
end;
end;
writeln('Арабские числа');
readln(n);
i := 0;
repeat
inc(i);
while (n >= b[i]) do
begin
n := n - b[i];
chisto := chisto + a[i];
end;
until n = 0;
writeln('Русские числа', chisto);
End.

```

№2. Такая система чаще всего используется там, где есть необходимость частого использования процедур десятичного ввода-вывода (электронные часы, калькуляторы).

№5 Program: пачка;
Var
i: array [1..1000] of integer;
b: array [1..1000] of string;
res: array [1..1000] of integer;
a: integer;
i, k, c: integer;
Begin
writeln('Введите количество экспериментов');

```
readln(a);
```

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

```
begin
```

```
  id[i]:=a;
```

```
  writeln('Введите значение параметра', i, ' экцеперемента');
```

```
  readln(b[i]);
```

```
  writeln('Введите результат', i, ' экцеперемента');
```

```
  readln(res[i]);
```

```
end;
```

```
for k:=1 to a do
```

```
  for c:=1 to a do
```

```
begin
```

```
  if res (mass[c] > res mas[c+1]) then
```

```
begin
```

```
  buf := mass[j];
```

```
  mass[j] := mass[j+1];
```

```
  mass[j+1] := buf;
```

```
  if := res[c] < res[a] then
```

```
begin
```

```
  i := res[c]; res[c] := res[a]; res[a] := i;
```

```
  s := b[c]; b[c] := b[a]; b[a] := s;
```

```
  assign i := id[c]; id[c] := id[a]; id[a] := i;
```

```
  for i:=1 to a do begin
```

```
    writeln('Наимр экцеперемента', id[i], ' Значение параметра', b[i], 'результат', res[i]);
```

```
  writeln;
```

```
end.
```

N 3.

Program qw;

Var.

a, b: integer;

i: longint;

c: longint;

cor: integer;

c1: integer;

begin.

readln(a);

readln(b);

cor := 0;

if $a < b$ then $c := a$ else if $b > a$ then $c := b$ else if $b = a$ then $c := a$;

for $i := 1$ to c do begin.

if $(a \bmod i = 0)$ and $(b \bmod i = 0)$ then $cor := cor + 1$;

end;

if $c = a$ then $c_1 := b$ else if $c = b$ then $c_1 := a$;

if $cor = 1$ then writel $n(c - c_1)$ else writel $n('He cyye cunbyem')$;

End.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ РАЗУМНЫЙ

ИМЯ ЮРИЙ

ОТЧЕСТВО АНДРЕЕВИЧ

Дата рождения 15.02.1997

Класс: 11

Предмет ИНФОРМАТИКА

Этап: ЗАКЛЮЧИТЕЛЬНЫЙ

Работа выполнена на 3 листах

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

Подпись участника олимпиады:

Чадн

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



```
5) #include <iostream> // C++
#include <cstdio>
#include <cmath>
#include <algorithm>
#include <string>
using namespace std;
int main()
{
    long long m, n, i, j;
    long long mas[100500];
    cin >> n >> m; // считываем значение размера массива
    for (i=0; i<(m*n); i++) /* зададим массив, в который
    { cin >> x; mas[i]=x; } добавим числа с меткой */
    sort(mas, mas+(m*n), greater<int>());
    /* отсортируем элементы массива в порядке убывания */
    for (j=0; j<n; j++) /* выведем через пробел числа в
    { for (i=0; i<m; i++) // порядке убывания до числа m,
    { cout << mas[i] << " "; } затем сменем на элемент
    cout << endl; }
    return 0; }
```

1) C++

```
#include <iostream>
#include <cstdio>
#include <cmath>
#include <algorithm>
using namespace std;
int main()
{
    long long i, n, j, answer=0;
    long long mas[1000000];
    mas[0]=0; mas[1]=1; cin >> n;
    for (i=2; i<999999; i++)
    { mas[i]=mas[i-1]+mas[i-2];
    if (mas[i]>n) { j=i-1; break; }
    /* выведем первые 2 элемента массива
```



значение n и i . Далее запишем массив mas числами Любоваги (каждое число равно сумме 2 предыдущих) и проверим будет ли число n больше или равно массива. Как только n станет меньше, приведем переменную j номер предыдущего n -го массива и оставшимся числам $*$

```
while (n > 0)
{ if (n >= mas[j])
{ answer = answer + pow(10, j);
  answer = answer / 10;
  n = n - mas[j]; }
  j = j - 1; }
cout << answer;
return 0; }
```

```
③ #include <iostream>
#include <cstdio>
#include <cmath>
#include <algorithm>
```

```
using namespace std;
int main ()
```

```
{ long long p, s, d, n, k, z, i, q;
```

```
cin >> p; q = p;
```

```
p = p - 1;
```

```
while (p % 2 == 0)
```

```
{ p = p / 2; z++; }
```

```
d = pow(2, z);
```

```
for (i = 0; i < q; i++)
```

```
{ for (s = 0; s < z - 1)
```

```
{ cin >> s; if (pow(i, d * pow(2, s)) != 1) && (pow(i, d) != 1)
```

```
cout << "составное"; r++; }
```

```
if (r != 0) cout << "составное";
```

```
else cout << "Вероятно простое";
```



```
return 0; }
```

```
(4) #include <iostream>
#include <cstdio>
#include <cmath>
#include <algorithm>
using namespace std;
int main()
{ int64 n, m;
cin >> n >> m;
cout << pow(n, m);
return 0; }
```

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7111

Ангарск 408
И-11-3

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Руденко

ИМЯ Данис

ОТЧЕСТВО Анатольевич

Дата рождения 16.02.1998

Класс: 11


Предмет информатика

Этап: заключительный

Работа выполнена на 3 листах

Дата выполнения работы: 03.03.2015
(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.



2. Двоично-десятичная СС применяется для возможности работы с цифрами числа, представленными в двоичной системе счисления.

```
3. var n, a : integer;
    function NOD(x, y : integer) : integer;
    begin
        if x < 0 then
            NOD := NOD(y mod x, x)
        else NOD := y;
        end;
    begin
        readln(n, a);
        if NOD(n, a) < 1 then
            writeln('не существует')
        else
            writeln((n - (n div a)) / (n mod a));
        end.
```

```
4. var n, i, j, c : integer;
    begin
        readln(n);
        if n = 1 then writeln(4)
        else if (n = 2) or (n = 3) then writeln(3)
        else
            begin
                c := 3; i := 1; j := 4;
                while n > (c + 4) do
                    begin
                        inc(i);
                        inc(j, 2);
                        c := c + j;
                    end;
                if ((n - (c + 4 - j + 1) + 1) mod (i + 1)) = 0 then
                    writeln(3)
                else writeln(2)
            end;
        end.
```



5. var a: array [1..1000] of integer;
b: array [1..1000] of string;
s: string; i, j: integer;
procedure QS(L, r: integer); {процед. быстрой сорт.};
var i, x, y, j: integer; s: string;
begin
i := L; j := r;
x := a[(L+r) div 2];
repeat
while (a[i] < x) do inc(i);
while (x < a[j]) do dec(j);
if (i <= j) then {no возврат.};
begin
y := a[i]; a[i] := a[j]; a[j] := y;
s := b[i]; b[i] := b[j]; b[j] := s;
inc(i); dec(j);
end;
until (i > j);
if (L < j) then QS(L, j);
if (i < r) then QS(i, r);
end;

begin
assign(f, 'input.txt'); reset(f);
while not eof(f) do {считыв. данных до конца файла};
begin
readln(s); s := s + ' ';
delete(s, 1, pos(' ', s));
inc(i);
b[i] := copy(s, 1, pos(' ', s) - 1);
delete(s, 1, pos(' ', s));
a[i] := strtoint(copy(s, 1, pos(' ', s) - 1));
end;
QS(1, i);
for j := 1 to i do
writeLn(b[j], ' ', a[j]); {вывод отсор. массива};
end;



6. var
a: array [1..1000, 1..1000] of integer;
m, n, i, j, t, d, s1, s2 : integer;
begin
 readln (m, n);
 for i := 1 to m do
 for j := 1 to n do
 readln (a[i, j]);
 for i := 1 to m do
 for j := 1 to n do
 if a[i, j] < 0 then
 break;
 for t := i to m do
 for d := j+1 to j+2 do
 if d = j+1 then s1 := s1 + a[t, d];
 else s2 := s2 + a[t, d];
 if s1 > s2 then writeln ('Temp')
 else if s1 < s2 then writeln ('Dama')
 else writeln ('Hurra!');
end.

1. var
 function analys (N: integer): string;
 var
 begin
 s := inttostr (N);
 for i := 1 to length (s) do
 begin
 k := length (s) + i - 1;
 if (s[i] >= 1) and (s[k] <= 3) then
 for j := 1 to n do
 if k = 1 then
 s := s + 'I'
 else if k = 2 then s := s + 'X'
 else if k = 3 then s := s + 'C';
 if s[i] = 4 then
 if k = 1 then
 s := s + 'V'
 else if k = 2 then
 s := s + 'L'
 else if k = 3 then s := s + 'D';

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ СТАРОДУБЦЕВ

ИМЯ АЛЕКСЕЙ

ОТЧЕСТВО АМИТРИЕВИЧ

Дата рождения 16.01.1997

Класс: 11

Предмет информатика

Этап: заключительный

Работа выполнена на 4 листах

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

Подпись участника олимпиады:

Амар

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 C++

```
#include <iostream>
#include <cstdio>
#include <algorithm>

using namespace std;
int main()
{
    long long s, n, a[1000000]; // описываем переменные
    int i;
freopen
    cin >> n; // считываем число в десятичной сс
    s = 0;
    a[0] = 1; a[1] = 1; // первые два числа
                        // в последовательности
                        // Фибоначчи = 1
    // Далее запалим массив числами Фибоначчи
    // до тех пор, пока последний элемент последовательности
    // станет больше либо равен данному числу
    i = 2;
    while (s < n)
    {
        a[i] = a[i-1] + a[i-2]; s = a[i]; i++;
    } i--;
    // Таким образом, у нас a[i] будет максимальным
    // числом последовательности
    // Далее, чтобы иметь тот вариант, если последний
    // элемент последовательности > данное число, чтобы
    // внагл не было ведущих нулей проинициализируем
    // массив:
    if (a[i] > n) i--; // i будет = количеству символов
                        // в числе Фибоначчи сс.
    // теперь начинаем выбирать значения 0 и 1
    go:
    while (n >= 1)
    {
        if (n % 2 == 0)
        {
            cout << "0 ";
            n /= 2;
        }
        else
        {
            cout << "1 ";
            n = (n-1)/2;
        }
    }
    cout << "\n";
}
```



```

{ if (a[i] ≤ n) { n = n - a[i]; count << 1; };
  else count << 0; i--; }

```

В случае если i у нас оказалась положительной, нам необходимо вывести еще столько 0, сколько $= i$

```

while (i > 0)
{ count << 0; i--; }

```

Полученная выверенная последовательность цифр и будет выводиться в обратном порядке системы счисления

```
return 0;
```

```
}
```

N5 C++

```

#include <iostream>
#include <cstdio>
#include <algorithm>

```

// подключаем нужные библиотеки

```
using namespace std;
int main()
```

```
{ long long n, m, a[100000000]; // объявляем переменные
```

```
cin >> m >> n; // считываем размеры страницы, где m - длина строки, n - длина строки
```

```
for (i = 0; i < m * n; i++)
```

```
cin >> a[i]; // считываем все числа на поле
```

```
sort(a, a + m * n); // сортируем все числа по возрастанию
```

```
for (i = m * n - 1; i ≥ 0; i--)
```

```
{ cout << a[i] << " "; if (m * n - i % m == 0) cout << endl; }
```

// Выводим числа по убыванию и в конце каждой строки переходим



```

    return 0; }
}

```

```

// 3 C++
#include <iostream>
#include <cstdio>
#include <cmath>
#include <algorithm>

using namespace std;

int main()
{
    long long p, d, s, r, t, k, a;
    cin >> p;
    if (p % 2 == 0) cout << "Составное";
    else
    {
        d = p - 1; while (d % 2 == 0)
        {
            d = d / 2; s++; } // найдем d и s таким способом
        while (pow(2, t) <= p)
        {
            t++; // найдем t
            bool l[0]; bool l[t+1]; bool l[t+1]; bool l[t];
            for (a = 1; a <= t; a++)
            {
                if if (pow(a, d) % p == 1) l[a] = 1; // если выполняется первое условие
                else for (r = 0; r <= s - 1; r++)
                {
                    k = pow(2, r);
                    if (pow(-1, k * d) % p == -1) l[a] = 1; }
            }
            // Там мы проверяли второе условие, если хоть одно из условий выполняется при каком-то a, то l[a] = 1;
            for (a = 1; a <= t; a++)
            {
                if (l[a] == 0) { cout << "Составное"; return 0; }
                cout << "вероятно простое"; return 0; }
            // Там если у нас хоть у одного элемента a не выполнялось
    }
}

```



Или одно из условий, то программа выберет
"оставшие", а иначе "вероятно простое"

№6 с++

```
#include <iostream>
#include <cstdio>
#include <algorithm>
#include <cmath>

using namespace std;

int main()
{
    long long M, N, K, L, P, TMC[1000000];
    int i;
    cin >> M >> N >> K >> L >> P; // считаем
    for (i = M; i <= N; i++) TMC[i] = 0;
    // считаем все числа long long a[10000000000];
    for (int i = 1; i <= K * L * 1000000 / 2; i++)
    {
        cin >> a[i];
        // a[i]
        TMC[a[i]]++;
    }
    cout << "найдем все простые с << i << " << N <<
    << " << TMC[i];
    return 0;
}
```


Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7107

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ

Тугай

ИМЯ

Вероника

ОТЧЕСТВО

Александровна

Дата
рождения

14.07.1999

Класс: 10

Предмет

информатика

Этап: II заключительный

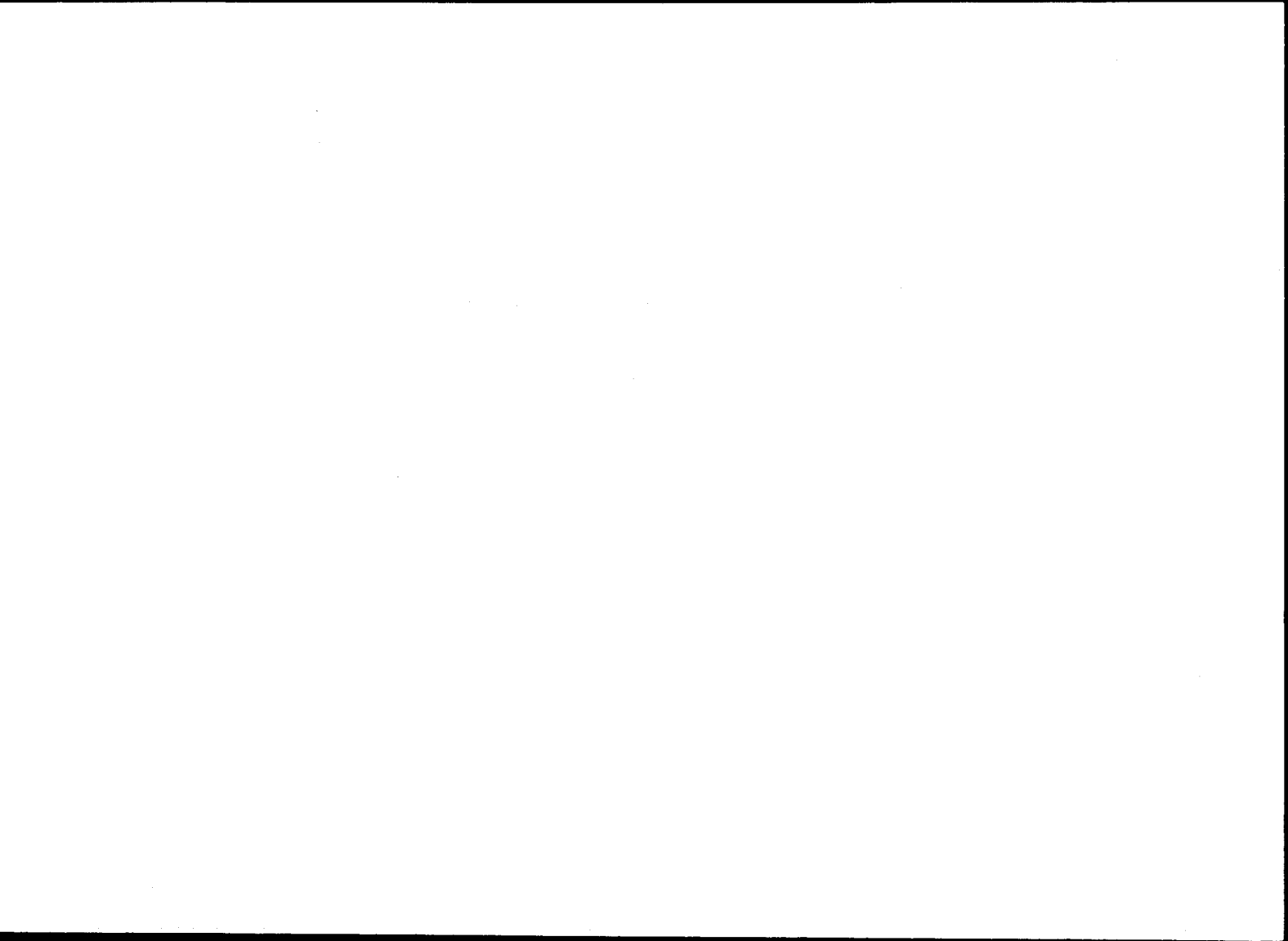
Работа выполнена на _____ листах

Дата выполнения работы: 20.03.2015
(число, месяц, год)

Подпись участника олимпиады:

Вероника

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





N 1.

1. переводим число из восьмеричной системы в десятичную.
 (обозначим число в 10-ной с. за a).

$$2. m_1 = a \operatorname{div} 1000;$$

$$d_1 = a \operatorname{mod} 1000;$$

$$d_1 = d_1 \operatorname{div} 500;$$

$$c_1 = d_1 \operatorname{mod} 500;$$

$$c_1 = c_1 \operatorname{div} 100;$$

$$l_1 = c_1 \operatorname{mod} 100;$$

$$l_1 = l_1 \operatorname{div} 50;$$

$$x_1 = l_1 \operatorname{mod} 50;$$

$$x_1 = x_1 \operatorname{div} 10;$$

$$v_1 = x_1 \operatorname{mod} 10;$$

$$v_1 = v_1 \operatorname{div} 5;$$

$$i_1 = v_1 \operatorname{mod} 5;$$

3. Теперь выводим римские числа:

M пишем m раз

D пишем d раз

C пишем c раз

L пишем l раз

X пишем x раз

V пишем v раз

I пишем i раз.

Готово!

Примечание: если п.1. недостаточно понятен, и выполнить не знает, как переводить из восьмеричной с. в десятичную, то это пишем над каждой цифрой числа в восьмеричной с. от 0 и по возрастанию, двигаемся справа налево.

пример:

арс $\begin{matrix} 2 & 1 & 0 \\ 3 & 3 & 7 \end{matrix}$

а) умножаем каждую цифру восьмеричного числа на $8^{\text{степень}}$ числа над ней и складываем их, получившееся число и есть a_0 .

пример:

$$a_{10} = 3 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0 = 223$$

$$a_8 = 337$$

$$a_{10} = 223$$

N2

Применение десятично-двоичной с.с. чисел удобно в том случае:

1. числа быстрее вводятся на индикацию.
2. если при переводе в десятичную с.с. и наоборот требуется легкость.
3. при использовании 2-10-ой с.с. проще умножить ^{и увеличить} на 10 и округлить

N4

представьте десятично-двоичную таблицу

N в обратном - мин. кол-во ячеек

$$\begin{array}{r} 1-4 \\ 2-3 \\ 3-3 \\ \hline 4-2 \end{array} 1$$

теперь разобьем ее на сектора так, чтобы в каждом была одна тройка, потому что это тройковая квадрат.

$$\begin{array}{r} 5-3 \\ 6-2 \\ \hline 7-3 \\ 8-2 \\ \hline 9-2 \end{array} 1$$

Наблюдается закономерность количества двоек в секторах 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6 и т.д.

$$\begin{array}{r} 10-3 \\ 11-2 \\ \hline 12-2 \end{array} 2$$

применим кол-во двоек на 1 меньше чем кол-во ячеек в секторе.

$$\begin{array}{r} 13-3 \\ 14-2 \\ \hline 15-2 \\ 16-2 \\ \hline 17-3 \\ 18-2 \\ \hline 19-2 \\ 20-2 \\ \hline 21-3 \\ 22-2 \\ \hline 23-2 \\ 24-2 \\ \hline 25-2 \\ 26-3 \\ \hline 27-2 \\ 28-2 \\ \hline 29-2 \\ 30-2 \end{array} 2$$

$$\begin{array}{cccccc} \underline{1, 1} & \underline{2, 2} & \underline{3, 3} & \underline{4, 4} & \underline{5, 5} & \\ 2 & 4 & 6 & 8 & 10 & \end{array}$$

$$\begin{array}{r} 13-3 \\ 14-2 \\ \hline 15-2 \\ 16-2 \\ \hline 17-3 \\ 18-2 \\ \hline 19-2 \\ 20-2 \\ \hline 21-3 \\ 22-2 \\ \hline 23-2 \\ 24-2 \\ \hline 25-2 \\ 26-3 \\ \hline 27-2 \\ 28-2 \\ \hline 29-2 \\ 30-2 \end{array} 3$$

Чтобы определить количество рядов надо, наоборот, вычитать от N числа 2, 4, 6, 8, 10, 12 и т.д. до тех пор пока $n \leq 0$.

пример: пусть $n = 24$

$$24 - 2 = 22; 22 - 4 = 18; 18 - 6 = 12; 12 - 8 = 4; 4 - 10 = -6$$

$$\begin{array}{r} 17-3 \\ 18-2 \\ \hline 19-2 \\ 20-2 \\ \hline 21-3 \\ 22-2 \\ \hline 23-2 \\ 24-2 \\ \hline 25-2 \\ 26-3 \\ \hline 27-2 \\ 28-2 \\ \hline 29-2 \\ 30-2 \end{array} 3$$

как только $n \leq 0$ мы берем предпоследнюю отнятую цифру (т.е. ту после которой n было минимально но больше нуля) для данного примера это 8.

$$\begin{array}{r} 24-2 \\ 25-2 \\ \hline 26-3 \\ 27-2 \\ \hline 28-2 \\ 29-2 \\ \hline 30-2 \end{array} 4$$

теперь просто прибавляем к этой цифре 3. получившееся число ...

$$\begin{array}{r} 27-2 \\ 28-2 \\ \hline 29-2 \\ 30-2 \end{array} 4$$

для 24 это 11



1. заполнить массив A - где Петра, B - где Данила
если страница M N M

$s1 := 0;$
 $s2 := 0;$

For $i = 1$ to N do begin

 read($a[i]$, $b[i]$)
 $s1 := s1 + a[i];$
 $s2 := s2 + b[i];$

end;

If $s1 < s2$ then
 write ln('Петр');

If $s2 < s1$ then
 write ('Данил');

end.

если страница M N M по номеру строки по месту
 $i := 1$ to N почитать $i := 1$ to M

NS.

1. Следует создать массив и заполнить его значениями результатов.

2. где i от 1 до N (где N - это N^0 п/п.) сравнивать

$a[i]$ и $a[i+1]$

$a[i]$ и $a[i+2]$

$a[i]$ и $a[i+3]$

и т.д. до:

$a[i]$ и $a[N]$

если $a[i]$ меньше то нужно переписать строку
второй результат равен $a[i]$ и не строки в результе-
том которой сравниваем $a[i]$

given base с любым значением и повторить все алгоритмы с начала

$$a \cdot b \bmod n = 1 \bmod n ; (a \cdot b - 1) \bmod n = 0$$

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7113

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ЦВЕТКОВ

ИМЯ Николай

ОТЧЕСТВО Викторович

Дата рождения 28.08.1997

Класс: 11

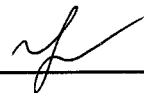
Предмет ИНФОРМАТИКА

Этап: Заключительный

Работа выполнена на 04 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.



```
(N) #include <iostream>
#include <cstdio>
#include <cmath>
```

// язык C++

```
using namespace std;
```

```
int main ()
```

```
{
```

```
long long n, a[1000000], i, b=0, z=0;
```

```
cin >> n; // считываем начальное число
```

```
a[0]=1; a[1]=1; // задаём первые элементы массива
```

```
for (i=2; i<999999; i++)
```

```
{ a[i] = a[i-1] + a[i-2];
```

```
if (a[i] > n) { b=i-1; break; }
```

```
}
```

```
/* заполнили массив a[1000000] числами
```

```
Фибоначчи до того числа, когда a[i] станет  
больше, чем n, и присвоили переменной b  
порядковый номер числа Фибоначчи, которое  
стоит перед n */
```

```
while (n > 0)
```

```
{ if (n >= a[b]) { z += pow(10, b-1); n -= a[b]; }
```

```
b--;
```

```
} // z - перевёрнутое число
```

```
cout << z;
```

```
return 0;
```

```
}
```




```
(NS) #include <iostream> // язык C++
#include <cstdio>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>

using namespace std;

int main ()
{
    long long m, n, i, j, a[1000000], z=1;
    cin >> m >> n;
    cin >> m >> n; // читаем размеры
    for (i=0; i<m*n; i++)
        cin >> a[i]; /* заполним массива a[1000000] всеми
        числами, которые у нас есть */
    sort (a, a + (m*n), greater <int > ()); /* отсортировали
    массив в порядке убывания */
    for (j=0; j<m; j++)
    {
        for (i=n*(z-1); i<n*z; i++)
            cout << a[i] << " ";
        cout << '\n';
        z++;
    } /* вывели числа по строкам в
    порядке убывания */
    return 0;
}
```



(N4) Язык: C++

Так как нет ограничений по времени и размеру, то возможно следующее решение для задачи

```
#include <iostream>
#include <cmath>
#include <cstdio>
```

```
using namespace std;
```

```
int main ()
```

```
{ int int64 m, n;
```

```
cin >> m >> m; cin >> m >> n;
```

```
cout << pow(n, m);
```

```
return 0;
```

```
}
```

(N3) Язык: C++

```
#include <iostream>
```

```
#include <cmath>
```

```
#include <cstdio>
```

```
#include <algorithm>
```

```
#include <vector>
```

```
#include <string>
```

```
using namespace std;
```

```
int main ()
```

```
{ long long s=0, d=0, p, a, r
  cin >> p;
```



```
d = p - 1;
while (d % 2 == 0)
{
    d /= 2;
    s++;
} // найдем d и s

for (a = 1; a <= p - 1; a++)
{
    for (r = 0; r <= s - 1; r++)
    {
for if ((pow(a, d * pow(2, r)) != -1 % p) &&
        (pow(a, d) != 1 % p)) { cout << "составное"; return 0; }
    }
} // *если условие не выполняется то пишем*
// *"составное" и выходим из программы*
cout << "Вероятно простое";

return 0;
}
```

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7101

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ШАРОКИН

ИМЯ Владислав

ОТЧЕСТВО Викторович

Дата рождения 17.06.1998

Класс: 10

Предмет Информатика

Этап: Заключительный

Работа выполнена на 2 листах

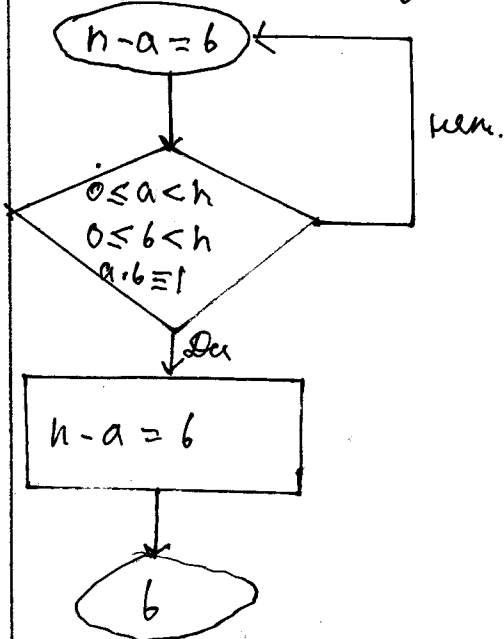
Дата выполнения работы: 03.03.15
(число, месяц, год)

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.



№3. Нахождение мультипликативно обратного шифра по заданному значению шифра. n -модуль; a -число шифра; b -результат.



$$5 - 3 = 2$$



2 мультипликативно обратное шифра.

№5 1) Нарисовать таблицу первая колонка шифр; вторая значение параметра, третья результат.

2) Пользуясь данными экспериментов заполнить таблицу, для этого ищем самое большое число в результатах, записываем его в таблицу и продолжаем по убыванию.

Пример.

№	шифр	рез.
1	А	3
2	Б	31
3	В	2
4	Г	6
5	Д	8
6	Е	19
7	Ж	20
8	З	19
9	И	43
10	К	1
11	А	50

Исходная табл.

№	шифр	рез.
1	А	50
2	Б	43
3	В	31
4	Г	20
5	Д	19
6	Е	19
7	Ж	8
8	З	6
9	И	3
10	К	2
11	К	1

исправленная таблица.

Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7103

И 10-5

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ Чербаков

ИМЯ Дмитрий

ОТЧЕСТВО Александрович

Дата рождения 12.11.1998

Класс: 10

Предмет Информатика

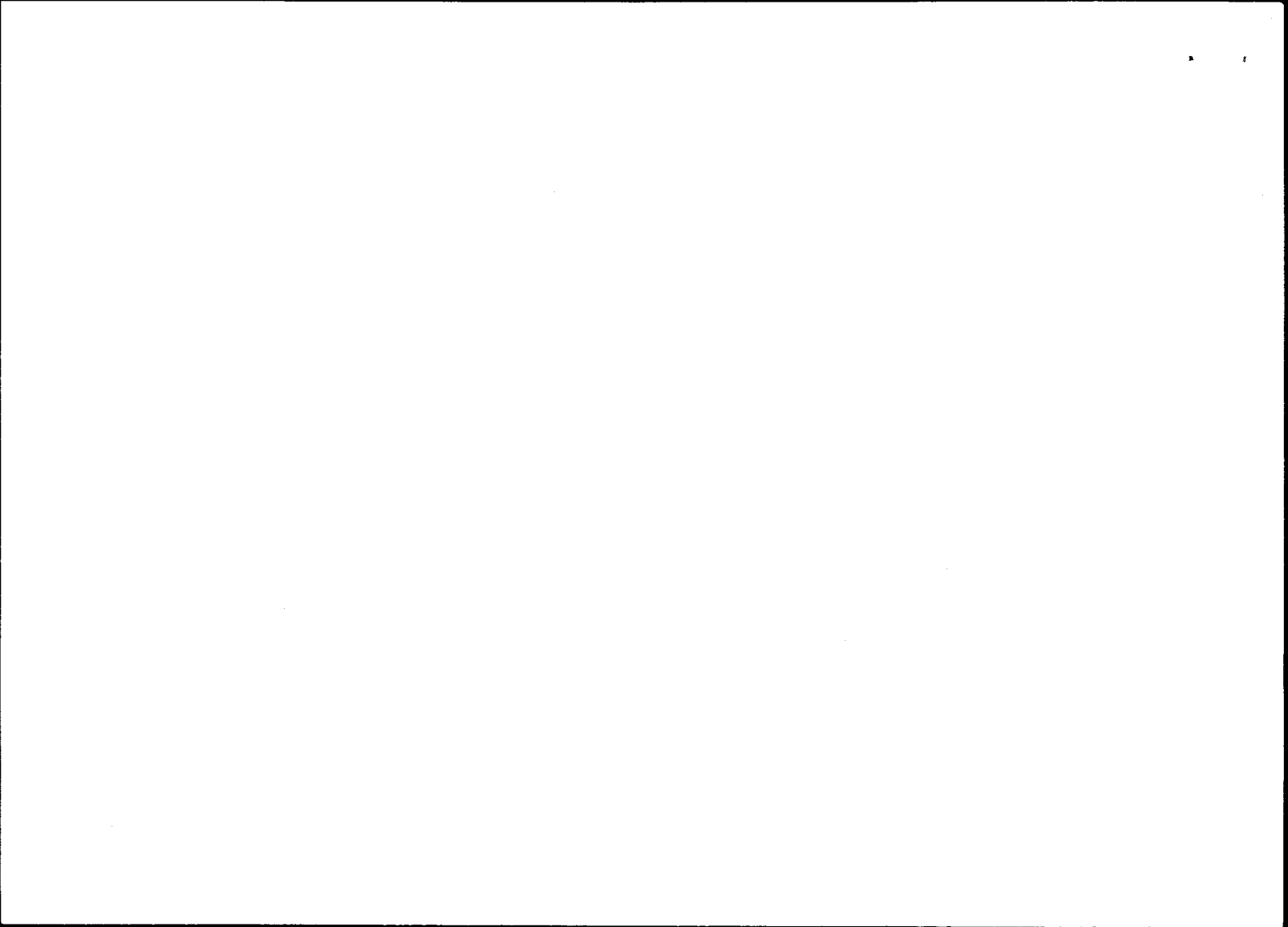
Этап: Заключительный

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.





①. Составим массив из чисел Фибоначчи, последний элемент которо-
рою не превышает данное N :

```
fib = [1, 2, 3]
```

```
do:
```

```
f = fib[-2] + fib[-1]
```

```
fib.append(f)
```

```
while f < N
```

Далее за каждый f , превыш с конца массива, превышающее N
мы добавляем в массив цифр 0, если f не превышает N , то
добавляем 1, а из N вычитаем f . до тех пор, пока N не рав-
но 0 или мы не дойдем до конца массива fib

```
num = []
```

```
x = 0
```

```
for f in reversed(fib): # развернем fib и пройдем по всем эл-мам
```

```
if f > N:
```

```
    num[x] = 0
```

```
else:
```

```
    num[x] = 1
```

```
    N = N - f
```

```
    x = x + 1
```

В итоге у нас получится массив из цифр, которые мы, после-
довательно объединив, представим как $N + fib$.

③. 1) Найдем S и d при данном p :

```
d = p - 1
```

```
while d % 2 != 0: # пока d делится на 2
```

```
    d = d / 2 # делим d на 2
```

```
S = S + 1
```

```
# степень увеличивается. В конце получим S и d.
```

2) Выберем t из промежутка $[1; p-1]$. $\log_2 p$ - нечетный.

```
t = floor(log(p, 2)) # округлим t, t, k. log2 p - нечетный.
```

```
for x in range(t): # проведем t операций цифр
```

```
    a.append(random.floor
```

```
    a.append(ceil(random() * (p-1))) # Возьмем случайную долю от
```

```
p-1, округленную вверх.
```

3) Для каждого a проверим условия:

```
либо  $a^d \% p = 1$  &
```

```
либо  $a^{2^t} \% p = p-1$ 
```

t найдем перебором от меньшего 101 до $S-1$.

Если есть одно a не подходящее, то результат - составное.

Если все a прошли проверку, то p - вероятно простое

④. Сопоставим степени работать проще.
Если представим N в виде $a+b$, то $N^m = (a+b)^m = (a+b)^x$, где x -
наибольшая степень, и $(a+b)^x$ представим в виде многочлена в 1
степени. В резу-те этих манипуляций получается большой
множитель в маленькой степени, ~~что~~ который легче вычисля-
ется.

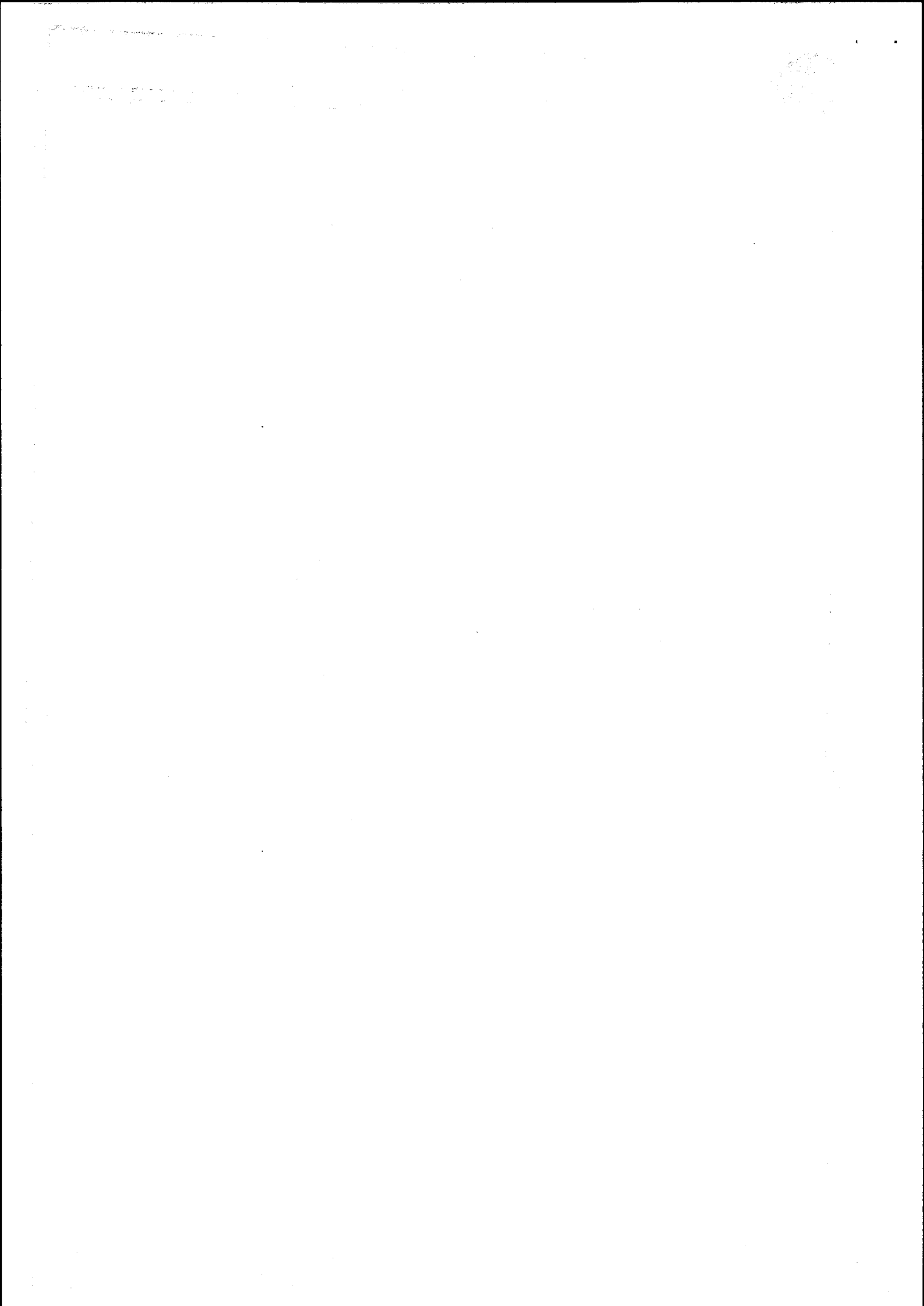
⑤. Числа можно упорядочить как слова в словарях, но в обратном
порядке алгоритма - сначала по длине чисел (от 1 до n),
а затем по разрядам ~~в каждой строке~~ в одном столбце, тем
же алгоритмом сортируем, если предыдущие разряды чисел
совпадают.



④. С операциями сложения работать проще.

Если представить N в виде $a+b$, то $N^M = (a+b)^M = (a+b)^x$, где x - небольшая степень. $(a+b)^x$ представим в виде многочлена. В итоге остается большой многочлен в маленькой степени, а это вычислить проще.

⑤. Числа сначала упорядочиваем по длине от 1 до N , а затем числа одинаковой длины и с одинаковыми предыдущими разрядами упорядочиваем по цифре от 0 до 9 в одной строке.



Олимпиада школьников «Надежда энергетики»

№ группы

Вариант № 7101

шифр

← Не заполнять
Заполняется
ответственным
работником

ФАМИЛИЯ ЯШИН

ИМЯ АРТЕМ

ОТЧЕСТВО ВЯЧЕСЛАВОВИЧ

Дата рождения 13.03.1998

Класс: 10


Предмет ИНФОРМАТИКА

Этап: II ЗАКЛЮЧИТЕЛЬНЫЙ

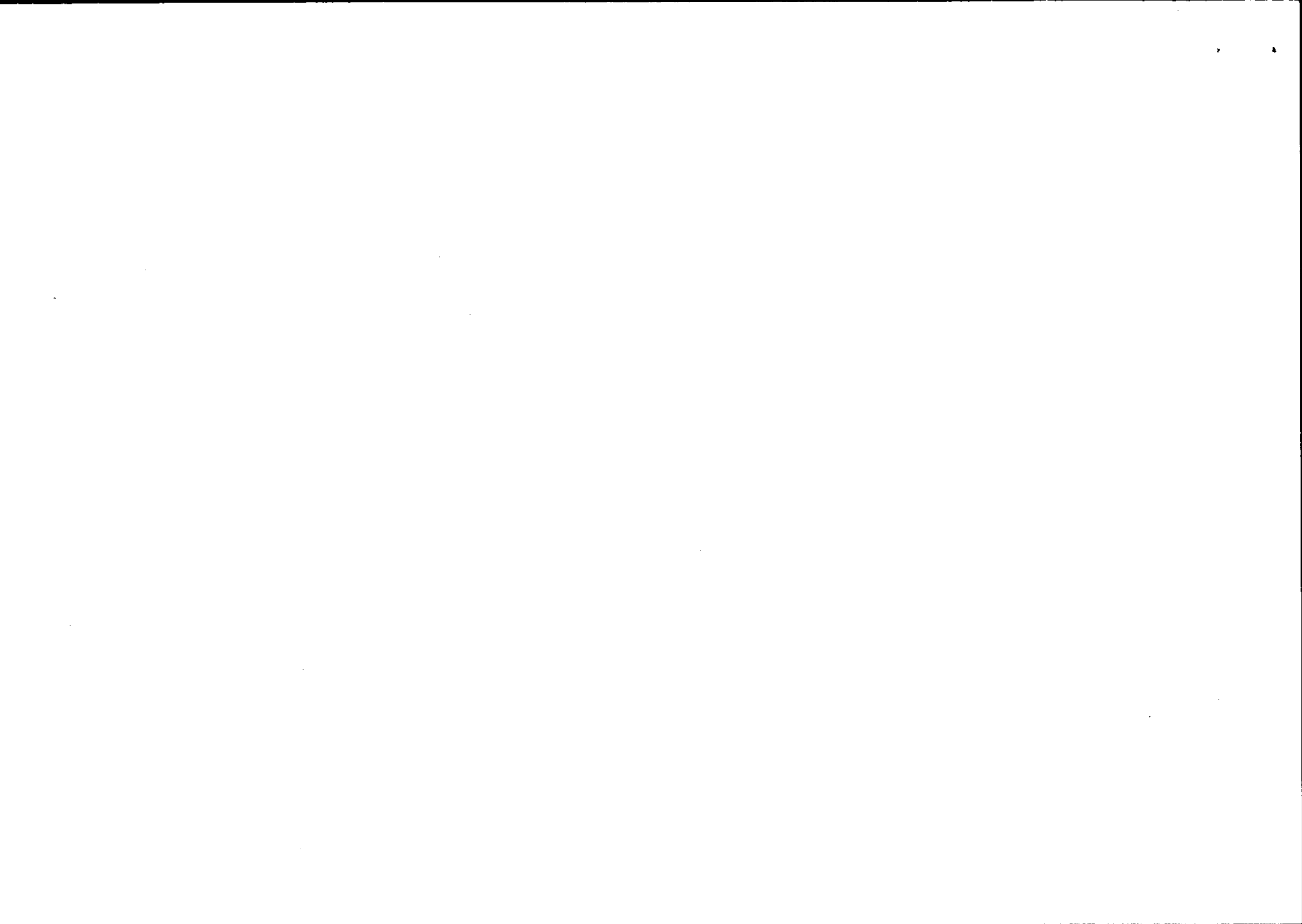
Работа выполнена на 2 листах

Дата выполнения работы: 20.03.2015
(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.





№5

Способ хранения данных;

Массив произвольного типа table.

Описание типа table:

id - целое беззнаковое число (считается 1!)

arg - действительное число.

res - целое число

sorted - текническая логическая переменная, по умолчанию false.

Алгоритм сортировки:

1) Возьмем массив с данными опытов - это a. Создадим массив целых беззнаковых чисел b, по длине равный a. В массив b мы будем сохранять значения id ячеек массива a по убыванию значений res.

2) Найти в массиве a ячейку с наибольшим показателем res и флажок sorted = false.

3) Когда ячейка найдена, сократить ее значение id в массиве b, а значение sorted выставить равным true.

4) Если массив b не заполнен, идти к пункту 2

5) Вывести значения arg и res массива a в порядке, заданном в массиве b. Надо учесть, что ячейки массива a нумруются от нуля, а значения id от 1. Следовательно, номер ячейки это id - 1.

№2

С помощью двоично-действительной записи легко получить цифру k любой цифре числа с помощью логической операции AND.

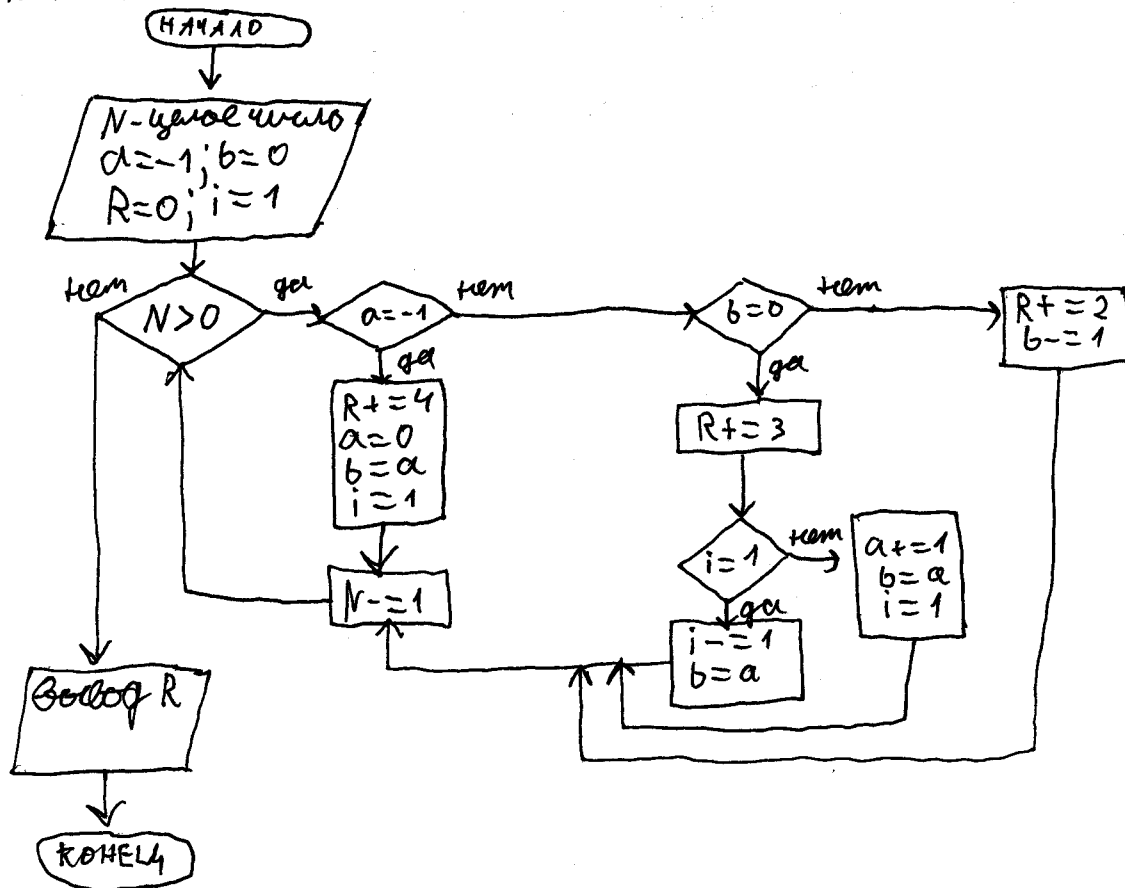
Тем десятичной же записи числа, чтобы получить цифру числа N под порядковым номером a (считая с нуля), надо сначала разделить N нацело на 10^a , а потом разделить остаток от деления на 10. Тем самым, если все это поместить в цикл (пока $N > 0$): $b = N \text{ mod } 10$; $N = N \text{ div } 10$ то цифры мы получим в обратном порядке.

Формула получения цифры b числа N под порядковым номером a (считая с нуля) в двоично десятичной записи:

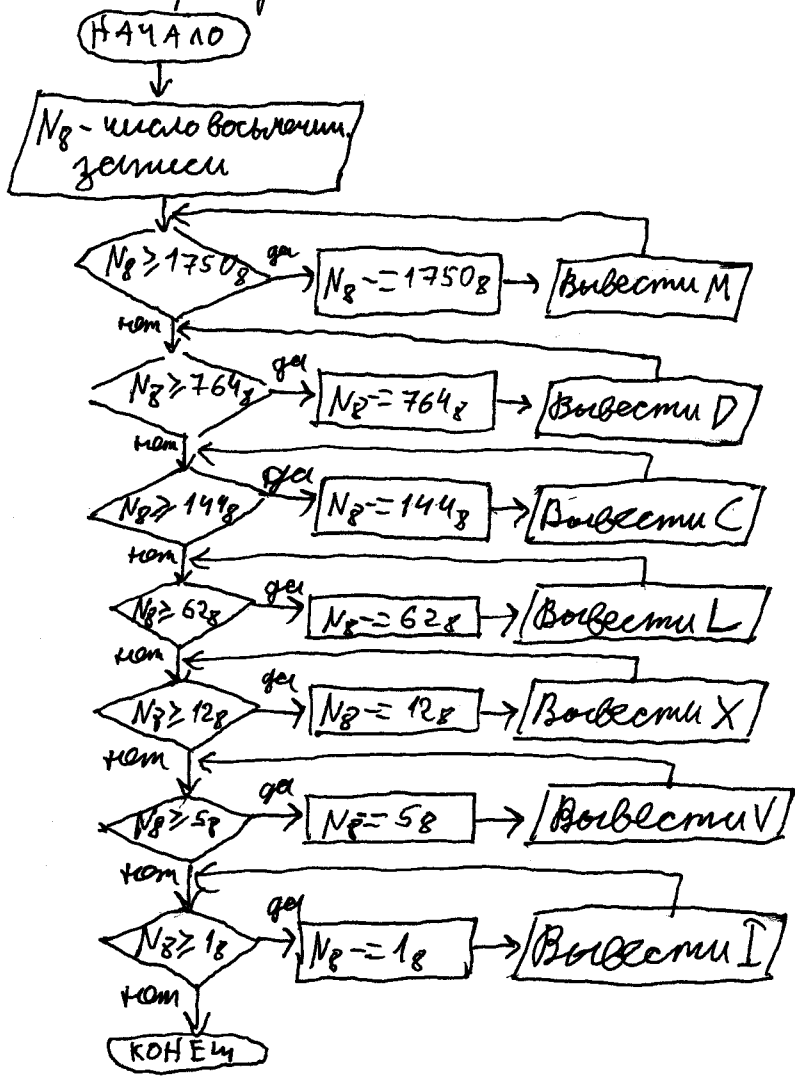
$$b = N \text{ AND } (0b1111 \ll (a \cdot 4))$$

см. на обороте ↵

N4



N1 "Трудный" алгоритм перевода:





№6

Минимальная ширина блока должна быть 20 ячеек, чтобы вместить названия столбцов (№ ПАРТИИ - 8, РЕЗ-Т1 - 6, РЕЗ-Т2 - 6)

Минимальная высота блока должна быть 2 ячейки, чтобы вместить названия столбцов и как минимум данные об одной партии.

Если $M < 20$ или $N < 2$, то входные данные неверны.

Если ширина блока больше 20, то подматрица все равно занимает 20 ячеек, а остальные ячейки игнорируются.

Считается, что, прочитав ячейку, указатель переходит на следующую. Если в строке ячейки кончились (указатель > 20), то происходит переход на следующую строку.

Если ячейка пуста, то четкие из не вернет 0.

Алгоритм чтения данных:

- 1) Пропустить 1-ую строку
- 2) Получить № ПАРТИИ (ПОДПРОГРАММА 1, аргумент: 8)
- 3) Получить РЕЗ-Т1 (ПОДПРОГРАММА 1, аргумент: 6)
- 4) Получить РЕЗ-Т2 (ПОДПРОГРАММА 1, аргумент: 6)
- 5) Прибавить к очкам Петра РЕЗ-Т1
- 6) Прибавить к очкам Даниила РЕЗ-Т2
- 7) Если у Петра очков меньше, чем у Даниила, то вывести "Петр", иначе "Даниил"

Подпрограмма 1:

Аргумент - длина поля.

Задача - прочитать число из ячеечного поля длины a .

Алгоритм:

- 1) $b = 0; c = 0$
- 2) Если $a > 0$, то идти на пункт 3, иначе на пункт 7
- 3) $b =$ число в ячейке
- 4) $a = a - 1$
- 5) $c = c + b \cdot 10^a$
- 6) идти на пункт 2
- 7) вернуть c .

см. на обороте 5

N3

