

Задача №4.

Володя решил переехать в другой город, но, к сожалению, он не может перевезти с собой все вещи, поэтому оставшиеся N предметов он решает продать и размещает объявления в интернете. На следующий день он проверил почту и обнаружил, что ему поступило M предложений о покупке, в каждом из которых покупатель готов приобрести K предметов, но только в случае, если Володя продаст все из списка сразу. Помогите Володе подсчитать количество предложений, с помощью которых он продаст наибольшее количество вещей.

Входные данные:

Первая строка: количество вещей N и количество предложений о покупке M через пробел;
Последующие строки: M предложений, каждое из которых содержит:
• Количество предметов в списке покупателя K ,
• Номера вещей из списка продаваемых.
Читать, что нумерация начинается с 1.

Выходные данные:

Целое число, соответствующее количеству предложений.

Пример:

Ввод:

4 3
2 1 2
2 2 3
2 3 4

Вывод:

2

Требования к оформлению задач по программированию:

- Программы должны быть написаны на одном из языков: C, C++, Pascal
- Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов

Ш



6360

1	2	3	4	Σ
6	9	0	18	33

заполняется жюри!

66

**ПИСЬМЕННАЯ РАБОТА УЧАСТНИКА
ОЛИМПИАДЫ ШКОЛЬНИКОВ СПбГУ
2016–2017**

заключительный этап

Предмет (комплекс предметов) Олимпиады ИНФОРМАТИКА (10-11 КЛАССЫ)

Город, в котором проводится Олимпиада _____

Дата _____

Вариант 01

Задача №1.

На планете «Крохотные крошки» в именах жителей не стоят две одинаковые буквы рядом. Необходимо написать оптимальный алгоритм генерации имен жителей планеты, если алфавит планеты состоит из 30 букв. Каждое имя состоит минимум из 3 букв, максимум из 10.

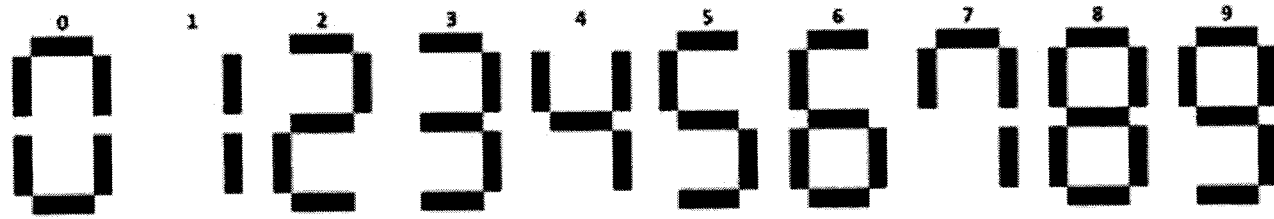
Требования к оформлению задач по программированию:

- Программы должны быть написаны на одном из языков: C, C++, Pascal
- Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов

Задача №2.

Цифровое табло, состоящее из семи-сегментных дисплеев, используется для вывода числовых значений. Вы можете *активировать выключенные* сегменты дисплея, но не можете *выключать* уже активные. Необходимо написать программу или алгоритм на языках C++, Pascal, Basic для определения максимального числа, которое может быть выведено на дисплей и не будет больше M.



Входные данные:

Целое число и ограничение M. Количество дисплеев равно количеству цифр во введенном числе.

Выходные данные:

Целое число. Количество цифр в конечном варианте должно быть равно количеству цифр в начальном.

Пример:

Ввод: 25 100

Вывод: 89

Требования к оформлению задач по программированию:

- 1) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 2) Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов

Задача №3.

Существует такая комната прямоугольной формы, с целочисленными длиной l и шириной w , для которых верно равенство $l = Aw + B$, где A и B – целочисленные постоянные. Количество единиц площади этой комнаты таково, что если прибавить к нему C , то результат будет делиться без остатка на простое число P . Найдите все возможные значения w .

Входные данные:

Первая строка содержит количество тестовых случаев, каждая следующая – тестовый случай с разделенными пробелом значениями A, B, C и P .

Выходные данные:

Для каждого случая с новой строки выводится результат решения, который содержит общее количество решений и полученные значения ширины в порядке возрастания. Все значения разделяются пробелом

Ограничения:

- $2 \leq P < 10^6$
- $0 < A < P$
- $0 \leq B$
- $C < P$

Пример:

Ввод:

Вывод:

2
1 1 0 2
1 2 2 3

2 0 1
0

Требования к оформлению задач по программированию:

- 3) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 1) Полностью оформленная задача должна содержать:
 - программу, выполняющую необходимые операции для всех допустимых данных;
 - операции с файлами входных и выходных данных или понятный пользователю интерфейс ввода исходных данных;
 - комментарии к тексту программы, облегчающие ее понимание.

Невыполнение вышеуказанных требований влечет за собой снижение получаемых за задачи баллов

Задача n1. (Pascal)

```
program n_1
var a, z, i : integer;
ar : array [1..10] of char;

procedure gen(g, pos : integer);
var j : integer;
begin
  ar[pos] := char(g); {добавляем символ с кодом g в результат}
  if pos >= 3 then {вывод имени >= 3 символа}
  begin
    for j := 1 to pos do
      write(ar[j]);
    writeln;
  end;
  if (pos < 10) then begin {рекурсия для вывода имен длиной больше pos}
    for j := a to z do
      if sign(g) <> i then
        gen(j, pos + 1);
  end;
end;
```

6

```
begin
  writeln('Программа выводит все возможные
  имена, состоящие из [3..10] букв
  букв букв алфавита в 30 букв');
  writeln('Введите код первой буквы алфавита
  (при условии, что буквы закодированы
  и занимают 30 подряд идущих ячеек);');
  read(a);
  z := a + 29; В writeln('Возможные имена:');
  for i := a to z do
    pr(i, 1);
end.
```

Примечание: т.к. вывести имена нужно в символах, которые неизвестны, предположим, что в памяти компьютера они расположены так же, как и латинские символы от a до z по порядку.

Задача №2. (Pascal);

```
program n.2;  
var a, b, i, j: integer;  
s1, s2;
```

Begin

```
WriteIn ('Введите данные через пробел');  
ReadIn (a, b);
```

```
s1 := a into str(a);
```

```
s2 := b into str(b);
```

```
IF length(s1s1) > length(s1s2) then b := '999999 ... 999999'  
max := 1;
```

```
For i := 1 to length(s1s1) do
```

```
Begin
```

```
case s1[i] OF
```

```
0: res := '0';
```

```
1: res := '0134789';
```

```
2: res := '28';
```

```
3: res := '389';
```

```
4: res := '489';
```

```
5: res := '5689';
```

```
6: res := '68';
```

```
7: res := 789 '0789';
```

```
8: res := '8';
```

```
9: res := '89';
```

```
end;
```

```
For j := 1 to length(res) do
```

```
For j := length(res) to 1 do
```

```
IF res[j] <= b[i] then begin
```

```
ans[i] := res[j]; break; end;
```

```
end;
```

```
WriteIn ('максимально возможное число');
```

```
For i := 1 to length(ans) do
```

```
write(ans[i]);
```

End.

Задача n 2 (Pascal)

```
program n-2  
var a, b, i, j: integer; integer  
s1, s2, ans, res: string;  
max: boolean;
```

```
begin WriteLn('Введите данные');  
ReadLn(a, b); { ввод данных }  
s1 := int intToStr(a); { преобразование данных в  
s2 := intToStr(b); { строки }  
IF length(s1) < length(s2) then max := 1  
else if length(s1) > length(s2) then begin  
for i := 1 to length(s1) - length(s2) do  
s1 := s1 + '0'; { приведение к одинаковой  
s2 := s2 + s1; { длине чисел }  
s2 := s1;  
end;
```

```
for i := 1 to length(s1) do  
begin
```

```
case s1[i] of  
0: res := '08';  
1: res := '0134789'; { данные о преобразованиях }  
2: res := '28';  
3: res := '389';  
4: res := '489';  
5: res := '5689';  
6: res := '68';  
7: res := '0789';  
8: res := '8';  
9: res := '89';  
end;
```

```
if max = 1 then
```

```
for ans[i] := res[length(res)];
```

```
else
```

```
for j := length(res) downto 1 do
```

```
begin if res[j] <= 6 if strToInt(res[j]) <= strToInt(s2[i]) then  
begin ans[i] := res[j]; break; end;
```

```
end.
```

```
WriteLn('ответ: ');
```

```
for i := 1 to length(ans) do
```

```
write(ans ans[i]);
```

```
End.
```

Задача 4. (Pascal)

~~const~~ const n = 10000 ; {ограничение на кол-во покупок}

type {предложение о покупке}

re = record;
col: integer;
a: array [1..n];

end;

var res: array [1..n] of boolean;
ar: array [1..n] of re.
f, o: text;

procedure pr(g, c : integer); {номер предложения и кол-во уже использованных предложений}

var er: boolean; ji: integer;

begin

for j:=1 to ar[g].col do

begin

if res[ar[g].a[j]] = 0 then

res[ar[g].a[j]] := 1

else er := 1;

end;

if er = 0 then begin

tov := tov - ar[g].col;

if c > max then

begin

max := c;

count := 1;

end

else if c = max then

count := count + 1;

if (a < m) and (tov > 0) then

begin

for j:=g to m do

pr(j, c+1);

end;

~~end~~ tov := tov + ~~ar[g].col~~; ar[g].col

end;

for j:=1 to ar[g].col do

begin

res[ar[g].a[j]] := 0;

end;

end;

begin

assign (f, 'input.txt');

assign (o, 'output.txt');

reset (f);

read (f, n, m);

{проверка на возможность покупки с учетом уже сделанных покупок}

{ = 0 - товар не продан; = 1 - продан }

{кол-во непроданных вещей}

{проверка на максимум покупок}

{рекурсивный перебор вариантов}

{возврат к данным до выполнения текущей подпрограммы}

```

max := 0;
For i := 1 to m do Begin
  Read (ar[i].col);
  For j := 1 to ar[i].col do
    Read (ar[i].a[j]);
  Readln;
  IF ar[i].col > max then begin
    max := ar[i].col;
    count := 1;
  end
  else
    IF ar[i].col = max then count := count + 1;
end;

```

```

For i := 1 to m do
  begin
    pr(i, 1);
  end;

```

```

Rewrite (o);
Write (o, count);
close (o);
End.

```

Примерание: массив обнулен по умолчанию
& Pascal (res)