

Задача №4.

На каждой клетке игрового поля размером $M \times N$ клеток можно поместить либо дозорную башню с лучниками, либо стену, либо оставить его пустым. Лучники на башне, находящейся по координатам (x, y) , могут атаковать другие башни по координатам $(x-2, y-2), (x-2, y_2), (x_2, y-2)$ и (x_2, y_2) , в том случае, когда они не разделены стеной.

На вход подается карта, на который отмечены свободные поля и поля с размещенными на них объектами. Ваша задача – разместить на карте как можно больше дополнительных башен, лучники на которых *не будут* атаковать друг друга.

Входные данные:

На вход может быть подано несколько тестов, каждый из которых имеет следующую структуру:

- Первая строка содержит два положительных целых числа M и N , разделенных пробелом
- Следующие M строк содержат N символов без пробелов, описывающих игровое поле, где
 - F – свободное поле
 - G – поле с башней
 - P – поле со стеной

Ввод данных считается законченным, когда M и N равны 0.

Пример:

Ввод:

FPFP
PFPP
GFGF
5 3
FPF
FFF
FGG
PFP
FPF
0 0

Вывод:

3
6

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



1	2	3	4	Σ
4	6	7	15	32

8238

заполняется жюри!

64

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

заключительный этап

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

Город, в котором проводится Олимпиада Санкт-Петербург

Дата 11.03.2017

Вариант 10

Задача №1.

На планете «Лютые лютики» имеется свой набор символов для использования в именах жителей. Алфавит состоит из $F > 10$ символов. Имена всех жителей планеты состоят из $D > 4$ букв. Правитель планеты требует, чтобы кто-нибудь смог создать таблицу всех возможных имен, при учете, что никакая буква в имени не повторяется более или равно 4 раза.

Входные данные: Размер алфавита, Алфавит строкой, длина имени

Выходные данные: строки имен (в файл или на экран).

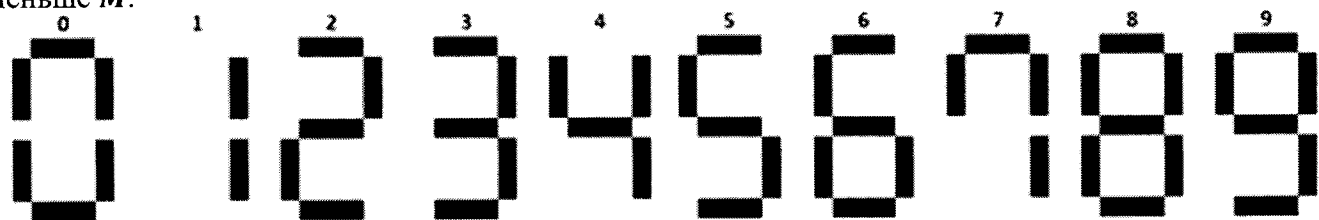
Требования к оформлению задач по программированию:

- 3) Программы должны быть написаны на одном из языков: $C, C++, Pascal$
- 4) 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 *деактивировать уже включенные* сегменты дисплея, но *включать* выключенные сегменты не можете. Необходимо написать программу или алгоритм на языках C, C++, Pascal для определения минимального числа, которое может быть выведено на дисплей и не будет меньше *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: 86 10

Вывод: 15

Требования к оформлению задач по программированию:

- 5) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 6) 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

```
Pascal
var a:array[0..9,0..9] of integer;
    i,k:integer; st,x,F,M,l1,l2:longint;
begin
    a[0,1]:=1; a[0,7]:=1; a[0,5]:=1; a[3,1]:=1; a[4,1]:=1; a[7,1]:=1;
    a[9,1]:=1; a[9,4]:=1; a[9,3]:=1; a[9,5]:=1; a[9,7]:=1;
    for i:=0 to 9 do if i<>8 then a[i,i]:=1;
    assign('input.txt',g); assign('output.txt',t); reset(g); rewrite(t);
    read(g,F,M); while x>0 do begin x:=x mod 10; k:=k+1; end;
    st:=1; for i:=1 to k-1 do st:=st*10;
    while st>0 do begin l1:=(F div st) mod 10; l2:=(M div st) mod 10;
    if l1>l2 then for i:=0 to 9 do if (a[l1,i]=1) and
    (a[l2,i]=1) then begin write(i); break; end
    else write(l1); st:=st/10; end; close(g); close(t); end.
```

Handwritten notes:
 {заполним матрицу цифрами (1) если число i может быть в числе j}
 {l1, l2 - беру по очереди числа из M и F и начинаю}
 {если l1 > l2 и это число может быть заменено по замене и вхожу из цикла}

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:

2
1 1 0 2
1 2 2 3

Вывод:

2 0 1
0

Требования к оформлению задач по программированию:

- 7) Программы должны быть написаны на одном из языков: C, C++, Pascal
- 8) 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

```
Pascal
var i,j,N:integer; A,B,C,P,k:integer; D,l1,x2:real;
    m,s,x1:longint; s:array[1..10] of longint;
begin
    assign('input.txt',g); assign('output.txt',t); reset(g); rewrite(t); readln(g,N);
    for i:=1 to N do begin readln(g,A,B,C,P);
    D:=(B*B)-(4*A*C); l1:=sqrt(D);
    if (l1<0) or (frac(l1)>0) then writeln(0) else begin
    x1:=abs((-B-l1)/(2*A)); x2:=abs((-B+l1)/(2*A));
    if ((A*x1*x1)+(B*x1)+C) mod P=0 then begin k:=k+1; s[k]:=x1; end;
    if ((A*x2*x2)+(B*x2)+C) mod P=0 then begin k:=k+1; s[k]:=x2; end;
    for i:=1 to k do for j:=1 to k do if s[i]>=s[j] then begin
    m:=s[j]; s[j]:=s[i]; s[i]:=m; end; write(t,k); for i:=1 to k do
    write(t,s[i]); writeln(t); k:=0; end; close(t); close(g); end.
```

Handwritten notes:
 {если l1 равно 0 или отрицательное или не целое число значит нет такой ширины}
 {если не целому корню под ним 2}
 {проверяю взаимность по условию (l1) mod P}
 если выполняется, то помещаю в массив
 {сортирую массив по убыванию}

Macro

```

1) Pascal
var
M, N: longint; i, j, k: integer;
c: char; a: array [1..10000, 1..10000] of integer;
begin
read(M, N); assign('input.txt', g); assign('output.txt', t); reset(g); rewrite(t);
write read(g, M, N);
while (M <> 0) and (N >= 0) do begin for i:=1 to 10000 do
for j:=1 to 10000 do a[i, j] := -1;
for i:=1 to M do
for j:=1 to N do begin
read(g, c);
if c = 'F' then a[i, j] := 0;
if c = 'P' then a[i, j] := 1;
if c = 'G' then a[i, j] := 2;
readln(g); end;
for i:=1 to M do
for j:=1 to N do begin
if (a[i, j] = 0) and ((a[i-2, j-2] <> 2) or (a[i-1, j-1] = 1)) and
((a[i-2, j+2] <> 2) or (a[i-1, j+1] = 1)) and ((a[i+2, j-2] <> 2) or (a[i+1, j-1] = 1)) and
and ((a[i+2, j+2] <> 2) or a[i+1, j+1] = 1)) then begin
a[i, j] := 2; k := k + 1; end;
writeln(k); readln(g, M, N); k := 0 k := 0; end; close(g); close(t); end;
end;

```

{ b массив переобнуем (-1)
 где zero код на кешах
 нуль } ~~и т.д.~~

{ запомним массив
 b коэффициентов с дробными }

{ проверю goes it на банни
 на нуле место и есть на
 строки }

(15)

```

2) Pascal
var F, D: longint; k: string;
i, j, M: integer;
S: array [1..10000] of integer; B: array [1..10000] of boolean;
begin
assign('input.txt', g); assign('output.txt', t); reset(g); rewrite(t);
read(g, F, k, D); M := 1;
while M <> 0 do begin for i:=1 to 10000 do begin S[i] := -1; B[i] := true;
for j:=1 to D do begin S[j] := 0; reset B[j] := true; end;
for j:=1 to M do begin if B[j] = true then begin
S[j] := S[j] + 1; if S[j] > F then begin S[j] := 0; B[j] := false; end; end;

```

{ b поначалу true
 массива переобнуем }
 { пока не станет
 больше F -
 затем переходим
 на следующий
 элемент } массива

(4)

else break; end; M = M + 1;

~~if~~ for i = 1 to N do write (k); write(m); end;

close(g); close(f); end.

{ belongs to memory
space (k) used
for memory
in eggs
representation }

