

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 которой 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.

Выходные данные:

Целые числа, соответствующие количеству дополнительных башен.

Пример:

Ввод:

```
PPFP
PEFP
GPGF
53
PBF
FFF
KGG
PFP
FPF
00
```

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



3777

1	2	3	4	Σ
6	8	10	5	29

58

заполняется жюри!

**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).

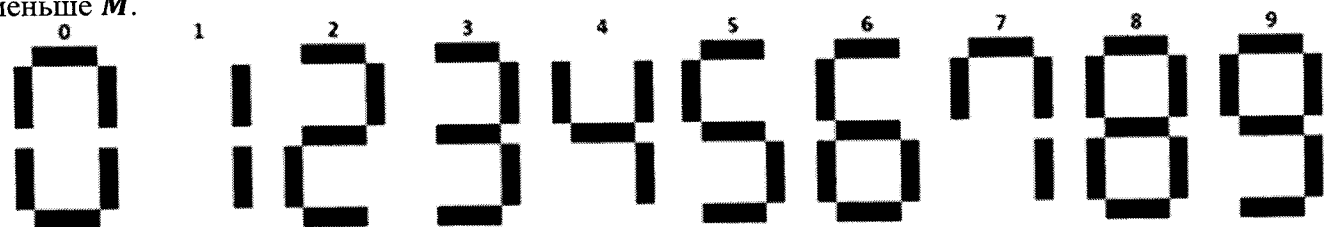
Требования к оформлению задач по программированию:

- 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

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

Мучалум.

Program n_1;

Var: KAZb, i, tofact1, tofact2, AllnameK, a, nomer, b, kname, prov: integer;

String: name[..];

Сирпана сомлоо илени

String: namep[..];

Сирпана га голеним бугаб в илени

String: AZb[..];

Сирпана аирпалум:

begin

writeln('Illegume kat-to cumlool & aipalume');

репар нолепна виленисирпана
сирпана аирпана

readln('KAZb');

if KAZb >= 10 then

begin

writeln('Illegume aipalume');

readln('AZb'); tofact1:= KAZb; tofact2:= KAZb;

лог ноленим га ноленим
м-ла голеним илени

KAZb:= length(AZb);

if KAZb < 10 then break;

ноленим иленисирпана сирпана аирпалум:

else

for i:=1 to KAZb do

илени га ноленим иленисирпана ноленим
ноленим сирпана

begin

AllnameK:= AllnameK + (tofact1 - (tofact2 - 2));

(иленисирпана ноленим иленисирпана)

tofact1:= tofact1 - 1;

if tofact1 = 0 then break;

end;

for a:=1 to AllnameK do

илени сирпана иленисирпана

begin

randomize;

nomer:= random(1, KAZb);

сирпана илени

name:= name + AZb[nomer]; kname:= length(name)

ноленим илени

for b:=1 to kname do

begin

if AZb[nomer] = name[kname] then

ноленим на ноленим ноленим
ноленим бугаб в илени

prov:= prov + 1;

end;

if prov = 4 and kname > 4 then

ноленим ноленим ноленим
4.

writeln(name);

end;

else writeln('aipalume girimo buni muniyem 10 cumlool');

end.

Сирпана ноленим иленисирпана ноленим. Сирпана ноленим иленисирпана

ноленим илени в ноленим илени, ноленим с ноленим иленисирпана ноленим

ноленим иленисирпана ноленим иленисирпана ноленим иленисирпана ноленим

ноленим иленисирпана ноленим иленисирпана ноленим иленисирпана ноленим



Program N-2;

var dlina1, dlina2, i

String: nachZnach[..];

String: M[..];

String: Min[..];

begin

writeln('legume yezha ruzo');

readln(nachZnach);

writeln('legume summarnaya ruzo opredeleniya');

readln(M);

dlina1 := length(nachZnach); dlina2 := length(M);

for i := 1 to dlina1 do

begin

if nachZnach[i] = 1 then

if 1 <= ~~M~~ M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 1;

if nachZnach[i] = 2 then

if 2 <= ~~M~~ M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 2;

if nachZnach[i] = 3 then

if 3 <= ~~M~~ M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 3;

if nachZnach[i] = 4 then

if 4 <= ~~M~~ M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 4;

if nachZnach[i] = 5 then

if 5 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 5;

if nachZnach[i] = 6 then

if 6 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 6;

if nachZnach[i] = 7 then

if 7 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 7;

if nachZnach[i] = 8 then

if 8 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 8;

if 1 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 1;

if 2 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 2;

if 3 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 3;

if 4 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 4;

if 5 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 5;

if 6 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 6;

if 7 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 7;

if 8 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 8;

if 9 <= M[i] then

if Min >= M then break; writeln(Min) else Min := Min + 9;

writeln(Min) else Min := Min + 1; if 2 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 2; if 3 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 3; if 4 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 4; if 5 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 5; if 6 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 6; if 7 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 7; if 8 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 8; if 9 <= M[i] then if Min >= M then break; writeln(Min) else Min := Min + 9; if nachZnach[i] = 0 and i = 1 then continue; if nachZnach[i] = 0 then ; ; ; if 0 = M[i] then Min := Min + 0;

end; writeln(Min); end.

zapros rychnogo i bezrychnogo slova dlya nachZnach i M.

zapros rychnogo i bezrychnogo slova dlya M i nachZnach to nach - opredeleniya

oboznacheniya i nachZnach i M dlya nachZnach i M.

zapros rychnogo i bezrychnogo slova dlya nachZnach i M.

zapros rychnogo i bezrychnogo slova dlya nachZnach i M.

oboznacheniya nachZnach i M dlya nachZnach i M.

2

oboznacheniya nachZnach i M dlya nachZnach i M.

Program n-4;
 var: M, N, i, q, dlina, PY, K

string: stline[.]; *запрос строки для строки*
 string: stAll[.]; *запрос строки строки для строки*

```

begin while (M=0) and (N=0) begin если не будем легким или программа будет  

  writeln('легкие запросы на M и N не могут быть'); работают  

  readln(M, N); используя значение M и N  

  for i:=0 to M do begin  

    writeln('запрос строки для строки');  

    read(stline);  

    if length(stline) = N then stAll := stAll + stline else writeln('неопределенный лог'); break;  

  end; PY := N * 2; каждый запрос строки для строки имеет длину от 2 до 0 Y.  

  dlina := length(stAll); выяснить значение длины строки со всеми данными на в.  

  for a:=0 to dlina do  

    begin  

if (stAll[i] = F) and (stAll[i+2] = F) or (stAll[i-2] = F) or (stAll[i+2] = 0) or (stAll[i-2] = 0)  

if (stAll[i] = F) and ((stAll[i+2] = F) or (stAll[i+2] = 0)) and ((stAll[i-2] = F) or  

(stAll[i-2] = 0)) and ((stAll[i+PY] = F) or (stAll[i+PY] = 0)) and (stAll[i-PY] = F) or  

(stAll[i-PY] = 0)) or (stAll[i+1] = F) or (stAll[i+1] = 0) and  

      if (stAll[i] = F) and ((stAll[i-2] = F) or (stAll[i-2] = 0) or (stAll[i-2] = P)) and  

      ((stAll[i+2] = F) or (stAll[i+2] = 0) or (stAll[i+1] = P)) and  

      поэтому возможно получение строки no 0X.  

      ((stAll[i+0Y] = F) or (stAll[i+0Y] = 0) or (stAll[i+N] = P)) and  

      ((stAll[i-0Y] = F) or (stAll[i-0Y] = 0) or (stAll[i-N] = P))  

      then k := k + 1  

    end; end;  

    writeln(k)  

  end. число повторений, не считая в себе и от - в зависимости от
```

Намного проще о строке не, преобразовать ее в одну строку
 и вывести результат вычисления на все определенное
 количество элементов, если нет, то вывести
 значение строки, иначе ее и вывести k и строку.
 После вычисления числа обратим строку на, чтобы k.

Умножение.

Program N-3:

Var A, b, c, p, k: integer;
var x1, x2: real; var deskr: real;

```

begin
  writeln('Введите начальные данные, включая количество переменных');
  readln(k); for i:=0 to k-1 do begin
    writeln('Введите начальные значения A, B, C и P');
    readln(A, b, c, p);
    if (p > 1000000) or (p < 2) then writeln('P должен быть между 2 и 1000000');
    if (c < p) then writeln('C должен быть меньше P');
    if (b > 0) then writeln('B должен быть больше 0');
    if (A <= 0) or (A >= p) then writeln('A должен быть между 0 и P');
  end;
  deskr := b*b - 4*A*c;
  if deskr < 0 then writeln('Для данных чисел решений нет');
  if deskr = 0 then begin
    x1 := (-b)/2*A;
    if mod(((A*x1+B)*x1+C)/P) = 0 then
      writeln('X1')
    else
      writeln('Для данных чисел решений нет');
  end;
  if deskr > 0 then begin
    x1 := (-b + sqrt(deskr))/2*A;
    x2 := (-b - sqrt(deskr))/2*A;
    if (mod(((A*x1+B)*x1+C)/P) = 0) then write('X1')
    else if (mod(((A*x2+B)*x2+C)/P) = 0) then write('X2')
    else
      writeln('Для данных чисел решений нет');
  end;
  if x1 > x2 then x1 := x2;
  if (mod(((A*x1+B)*x1+C)/P) = 0) then write('X1');
  if (mod(((A*x2+B)*x2+C)/P) = 0) then write('X2')
  else
    writeln('Для данных чисел решений нет');
end;

```

10

В данном случае блонн сформулирован по условиям, но с учетом, нормаль, лубри, пружин, упругих элементов и пружинистых элементов, не имеющих упругих элементов, тем не менее, не все свойства элементов упругих и лубри элементов с пружинистыми элементами.

