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.

Выходные данные:

Целые числа, соответствующие количеству дополнительных башен.

Пример:

Ввод:	***Вывод:***
	3
FPFP	6
PFPP	
GFGF	
5 3	
FPF	
FFF	
FGG	
PFP	
FPF	
0 0	

****

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

ЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

54

1

9633

1	2	3	4	Σ
0	9	0	18	237

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).

****

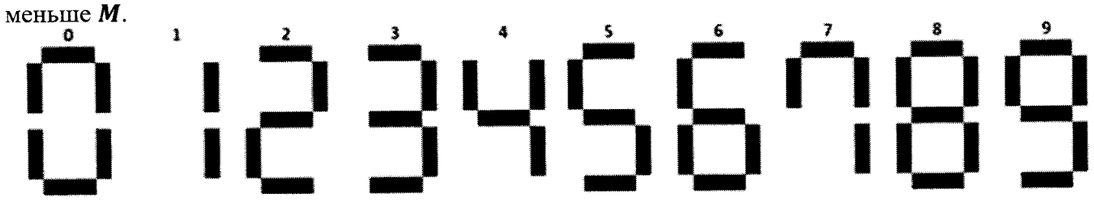
Требования к оформлению задач по программированию:

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:

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

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

Чистовик

Задача № 2

```
var  mas0 : array [1..3] of integer;
     mas3 : array [1..2] of integer;  // создадим массивы mas0-9,
     mas4 : array [1..2] of integer;  // которые будут содержать
     mas6 : array [1..2] of integer;  // возможные превращения чисел
     mas7 : array [1..2] of integer;  // на цифрах от 0 до 9
     mas8 : array [1..10] of integer;
     mas9 : array [1..6] of integer;  // соответственно
     mas1, mas2, mas5, M, N, i, p, i1 : integer;
     K, L : array [1..10] of integer;
begin
     writeln ('Введите число с таблю и ограничение');
     readln (N, M);  i := 1;
  while N > 0 do begin
                          // "раскидаем" цифры числа
        K[i] := N mod 10; // по ячейкам массива K
        N := N div 10;    // в обратном порядке
        i := i + 1;
  i := i - 1; p := 1;  end;
  while M > 0 do begin
                          // аналогично поступаем
        L[p] = M mod 10;  // с ограничением (в
        M := M div 10;    // массив L)
        p := p + 1;
        end;
  p := p - 1;   // заполним массивы с превращениями
  mas0[1] := 0; mas0[2] := 1; mas0[3] := 7; mas1[1] := 1; mas2 := 2;
  mas5 := 5; mas3[1] := 1; mas3[2] := 3; mas4[1] := 1; mas4[2] := 4;
  mas6[1] := 5; mas6[2] := 6; mas7[1] := 1; mas7[2] := 7; mas8[1] := 0;
  mas8[2] := 1; mas8[3] := 2; mas8[4] := 3; mas8[5] := 4; mas8[6] := 5;
  mas8[7] := 6; mas8[8] := 7; mas8[9] := 8; mas8[10] := 9; mas9[1] := 1;
  mas9[2] := 3; mas9[3] := 4;
  mas9[4] := 5; mas9[5] := 7; mas9[6] := 9;
  for i1 := 1 to i do begin if i1 ≤ p then begin
     if K[i1] = 0 then if mas0[1] ≥ L[i1] then K[i1] := mas0[1]
                       else if mas0[2] ≥ L[i1] then K[i1] := mas0[2]
                       else if mas0[3] ≥ L[i1] then K[i1] := mas0[3]
  else if K[i1] = 1 then K[i1] := mas1
   else if K[i1] = 2 then K[i1] := mas2
```

```
else if k[i1]=5 then k[i1]=mas5
else if k[i1]=3 then if mas3[1]≥L[i1] then k[i1]:=mas3[1]
                     else if mas3[2]≥L[i1] then k[i1]:=mas3[2];
~~else if mas~~
else if k[i1]=4 then if mas4[1]≥L[i1] then k[i1]:=mas4[1]
                     else if mas4[2]≥L[i1] then k[i1]:=mas4[2]
else if k[i1]=6 then if mas6[1]≥L[i1] then k[i1]:=mas6[1]
                     else if mas6[2]≥L[i1] then k[i1]:=mas6[2]
else if k[i1]=7 then if mas7[1]≥L[i1] then k[i1]:=mas7[1]
                     else if mas7[2]≥L[i1] then k[i1]:=mas7[2]
else if k[i1]=8 then k[i1]:=L[i1]
else if k[i1]=9 then if mas9[1]≥L[i1] then k[i1]:=mas9[1]
                     else if mas9[2]≥L[i1] then k[i1]:=mas9[2]
                     else if mas9[3]≥L[i1] then k[i1]:=mas9[3]
                     else if mas9[4]≥L[i1] then k[i1]:=mas9[4]
                     else if mas9[5]≥L[i1] then k[i1]:=mas9[5]
                     else if mas9[6]≥L[i1] then k[i1]:=mas9[6];
                     ~~else if~~
                     end
else   K[i1]:=0;
  end; // выше мы перебрали варианты превращений
writeln('Минимально возможное число');
while(i>0) do ~~write~~ begin
                write(K[i]); // выводим массив K
                i:=i-1;      // в обратном порядке
                end;         // без пробелов
end.
```

(9)

Чистовик

Задача 4

```
var P : array [1..1000000, 1..1000000] of char;
    M, N, i, j, k, l, t : integer;
begin
  read (M), readln(N); l:=0;
  while (M<>0) and (N<>0) do begin
      for i:=1 to M do
          for j:=1 to N do begin
              if j = N then readln (P[i,j])
                       else read (P[i,j]);
                   end;
      for i:=1 to M do
        for j:=1 to N do begin
            t:=0 ; k:=0;
            if P[i,j] = 'F' then begin
                if (i-2>0) and (j+2 ≤ N) then begin
                    t:=t+1;
                    if (P[i-2, j+2] <> 'G') or (P[i-1,j+1] = 'P') then
                        k:=k+1;
                    end;
            if (i+2 ≤ M) and (j+2 ≤ N) then begin
                t:=t+1;
                if (P[i+2, j+2] <> 'G') or (P[i+1,j+1] = 'P') then
                    k:=k+1;
                end;
            if (i+2 ≤ M) and (j-2 >0) then begin
                t:=t+1;
                if (P[i+2, j-2] <> 'G') or (P[i+1, j-1] = 'P') then
                    k:=k+1;
                end;
            if (i-2>0) and (j-2>0) then begin
                t:=t+1;
                if (P[i-2,j-2] <> 'G') or (P[i-1,j-1] = 'P') then k:=k+1;
                end;
```

```
        if k=t then l:=l+1;
        end;
        end;
        writeln(l);
        read(M); readln(N);
        end;
    end.
```

// комментарии к программе:
// сначала мы считываем координаты 1 поля, считываем
// поле, а после для каждой свободной ячейки
// проверяем возможность установки башни нулём
// оценивания всех простреливаемых лучниками
// позиций. Если там нет лучников, или между
// башней и позицией есть стена, то оцениваем
// позицию положительно. Если кол-во положительно
// оценённых позиций ровно числу имеющихся позиций
// (т.е. не выходящих за рамки поля, а их 4), то
// ставим башню в точку с координатами $(x_i; y_i)$