САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

1	2	3	4	Σ
4	8	ϕ	~~18~~ 15	~~25~~ 27

54
50

заполняется жюри!

2 3852

ПИСЬМЕННАЯ РАБОТА УЧАСТНИКА ОЛИМПИАДЫ ШКОЛЬНИКОВ СПбГУ 2016–2017

заключительный этап

Предмет (комплекс предметов) Олимпиады ИНФОРМАТИКА (8-9 КЛАССЫ)

Город, в котором проводится Олимпиада г. Владимир

Дата 11.02.2017

****

Вариант 01

****

Задача №1.

На вход подается строка с математическим выражением, необходимо написать программу или алгоритм на языках C, C++, Pascal, вычисляющую результат выражения. Выражение должно содержать однотипные знаки $(+,-)$, но не менее 5 знаков.

Входные данные: Строка с математическим выражением.
Выходные данные: Число-ответ.

Пример:

Ввод: 2+21-35+10-5+11 ***Вывод:*** 4

****

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 №4.

На прямой улице находится $n+1$ домов с номерами от 0 до n, расположенные в порядке возрастания друг за другом. Стало известно, что грабители планируют проникнуть в некоторые дома на этой улице, поэтому полиция выслала сразу несколько патрульных групп на предполагаемое место преступления. Каждый патруль может охранять сразу несколько домов с подряд идущими номерами, и дом может находиться под охраной сразу нескольких групп одновременно. Патрули также могут передвигаться по улице, при этом количество домов, которые они могут держать под наблюдением одновременно остается неизменным. Если грабитель проникнет в дом, то диспетчеру должно быть доложено количество групп, находившихся на месте преступления в момент проникновения.

Входные данные:

- Первая строка содержит числа n и m, разделенные пробелом.
- m – количество строк, каждая из которых обозначает определенное действие и может иметь один из следующих форматов:
  - ***P u v***
    Высылает на улицу патрульную группу, которая будет держать под наблюдением дома с u-го по v-й включительно. Каждая высланная группа получает свой порядковый номер, начиная с 1.
  - ***M l d***
    Диспетчер дает команду группе с номером l перейти на d домов. Если $d < 0$, то группа двигается в начало улицы (к нулевому дому), если $d > 0$ – в конец (к дому с номером n)
  - ***B x***
    Грабитель пытается проникнуть в дом x

Выходные данные:

При каждой попытке ограбления выводится целое число, показывающее количество патрульных групп, охраняющих дом.

Ограничения:

- $0 < n < 10^9$
- $0 < m \leq 250000$
- $0 \leq u \leq v \leq n$

Пример:

Ввод:	Вывод:
7 5 P 1 4 P 3 5 B 3 M 2 1 B 3	2 1

****

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 №2.

Необходимо написать на специальном языке набор команд для построения правильного 8-угольника со всеми его диагоналями.

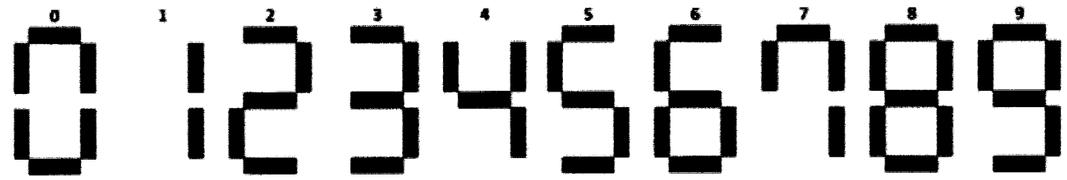
Доступны следующие команды:

$НЦ_x$, КЦ – начало и конец цикла, x – количество выполнений цикла;
НР, КР – начало рисования, конец рисования;
$ПВРТ\_ЛВ_{xx}$, $ПВРТ\_ПР_{xx}$– поворот влево или вправо на xx градусов на месте;
$ВП_{xx}$, $НЗ_{xx}$ – вперед или назад на xx шагов (1 шаг = 1 см);
ИСХ – команда возвращения робота в исходную позицию.
$ПРГ_x$ – команда для создания подпрограммы.

** Допускается создание подпрограмм. Тогда конечная программа будет выглядеть как набор подпрограмм с заданным порядком.*
*** Программа должна быть компактной и содержать циклические конструкции.*

Задача №3.

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, C++, Pascal 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:

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

Чистовик
Задача №1

```
program z1;
var s,s1: string;
    d,p,x,y,i,er,mn,pl : Longint;
    t: text;
begin
  assign(t,'in.txt');
  reset(t);
  read(t,s);
  close(t);
  mn:=pos('-',s);                    // mn - первый встречающийся минус
  pl:=pos('+',s);                    // pl - первый встречающийся плюс
  if (mn<pl) and (mn>0) then p:=mn   //находим первый знак в выражении
    else p:=pl;
  if p>1 then
     begin
       s1:=s[1];                     //записываем первое число
       for i:=2 to p-1 do
         s1:=s1+s[i];
       val(s1,y,er);
       delete(s,1,p-1);
     end
    else y:=0;
  while (length(s)>0) do
    begin
     if s[1]='-' then d:=-1          // проверяем, какой знак стоит перед числом
       else d:=1;
     delete(s,1,1);
     mn:=pos('-',s);
     pl:=pos('+',s);
     if (mn<pl) and mn>0 then p:=mn
       else if pl>0 then p:=pl
         else p:=length(s)+1;
     s1:=s[1];
     for i:=2 to p-1 do              // записываем число
       s1:=s1+s[i];
     val(s1,x,er);
     delete(s,1,p-1);
     if d=-1 then y:=y-x             // вычисляем
       else y:=y+x;
    end;
  assign(t,'out.txt');
  rewrite(t);
  writeln(t,y);
  close(t);
end.
```

(-1)

Задача №2

```
ClC-X
HLS8
 HP
 ВП2-√2
 kP
 НЗ2-√2
 ПВРТ_ЛВ22,5
 HP
 ВП1+√2
 kP
 НЗ1+√2
 ПВРТ_ЛВ22,5
 HP
 ВП√2+√2
 kP
 НЗ√2+√2
 ПВРТ_ЛВ22,5
 HP
 ВП1
 kP
 ПВРТ_ПР67,5
kLS
```

(8)

(Для вычисления расстояния, на которое должен двигаться исполнитель, я использовала теорему синусов и теорему косинусов, поэтому результат содержит корни)

Чистовик
Задача 24

```
program z4;
var n,m,s,s1,x,y,i,j:longint;
    p: array[1..250000,1..3] of longint;
    ot: array[1..250000] of longint;
    c: char;
    t: text;
begin
  assign(t,'in.txt');
  reset(t);
  read(t,n,m);
  for i:=1 to m do
   begin
    p[i,1]:=-1;
    p[i,3]:=-1
  s:=0;                    // s - количество патрульных групп
  s1:=0;                   // s1 - количество попыток ограбления
  for i:=1 to m do
    begin
      read(t,c);
      if (c='p') then
        begin
          s:=s+1;
          read(t,p[s,1], p[s,2])  // в массиве p хранятся номера 1-го и последнего домов
          p[s,3]:=p[s,2]-p[s,1]+1; // а также их количество для каждой патрульной группы
        end;
      if (c='m') then
        begin
          read(t, x, y);
          if (y>0) then
            begin
              p[x,2]:=n                    // при переходе группы меняем номера домов
              p[x,1]:=n-p[x,3]+1;
            end;
          if (y<0) then
            begin
              p[x,1]:=0;
              p[x,2]:=p[x,3]-1;
            end;
        end;
      if(c='b') then
        begin
          read(t,x);
          s1:=s1+1;
          ot[s1]:=0;                   // при попытке взлома смотрим, какие группы охраняют данный дом
          for j:=1 to s do             // и считаем их количество
            if (x>=p[j,1]) and (x<=p[j,2]) then ot[s1]:=ot[s1]+1; // сохраняем в массив
        end;
    end;
  close(t);
  assign(t,'out.txt');
  rewrite(t);
  for i:=1 to s1 do
     writeln(t, ot[i]);
   close(t);
end.
```

15