

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:

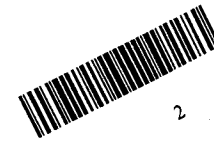
Ввод:	Вывод:
7 5	2
P 1 4	1
P 3 5	
B 3	
M 2 1	
B 3	

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

Шифр:



	2	3	4	Σ
6	7	12	2	27

заполняется жюри!

54

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

заключительный этап

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

Город, в котором проводится Олимпиада г. Челябинск

Дата 21.02.2017

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 №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.

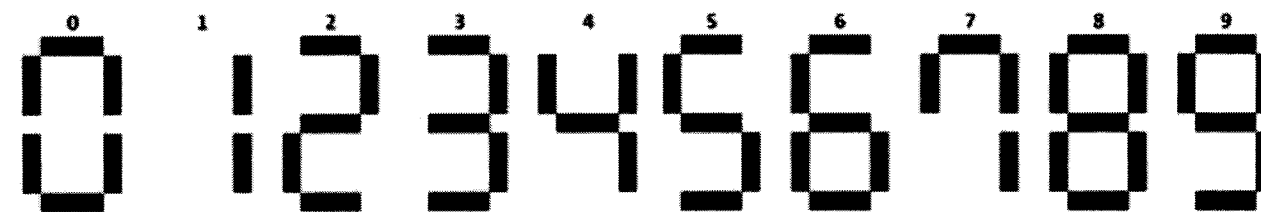
```

НЦ1
  НР
  НЦ8
    ВП1
    ПВРТ_ПР45
  КЦ
  НЦ8
    НЦ2
      ПВРТ_ПР22,5
      ВП1,5
      НЗ1,83
      ПВРТ_ПР22,5
      ВП2,4
      НЗ2,4
      ПВРТ_ПР22,5
      ВП2,6
      НЗ2,6
      ПВРТ_ПР22,5
      ВП2,4
      НЗ2,4
      ПВРТ_ПР22,5
      ВП1,83
      НЗ1,83
      ПВРТ_ПР22,5
      ВП1
      ПВРТ_ПР155
    КЦ
  КЦ
КР
КЦ
  
```

(7)

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

Язык: Pascal.

```
var i, sum, pl, mn, x : longint;
    f : boolean;
    s : string;
```

```
begin
  readln(s); // считываем задачу как строку
  i := 1;
  f := true;
  pl := 0; // положительные
  mn := 0; // отрицательные
  sum := 0; // конечный результат
  if s[1] = '-' then begin
    f := false;
    i := 2;
  end; // проверяем, нет ли перед 1 числом минуса

  while i <> length(s) do
  begin
    x := 0; // переменная, в которой будет лежать число
    while (s[i] <> '+') AND (s[i] <> '-') do // считываем каждое число
    begin
      x := x * 10;
      x := x + s[i];
      inc(i);
    end;
    if f = true then pl := pl + x // pl - переменная, в которой лежат положительные числа
    else mn := mn + x; // mn - переменная, в которой лежат отрицательные числа

    if s[i] = '+' then f := true
    else f := false; // решаем, куда положить следующее число

    inc(i);
  end;
  sum := pl - mn; // находим разницу между положительными и отрицательными числами
  writeln(sum); // выводим результат;
end.
```

6

Язык: Pascal

Задача №3

var: b, c, otv: array[1..1000] of integer;
i, j, k, M, x, ch, p, kl: longint;

```

begin
  read(ch, M);
  k:=0; kl:=1;
  x:=ch;
  while x<>0 do
  begin
    x:=x div 10;
    inc(k);
  end; // узнаем, сколько цифр в числе
  for i:=1 to k do
  begin
    b[i]:=ch mod 10;
    ch:=ch div 10;
  end; // собираем цифры числа в массив
  for i:=k downto 1 do

```

```

begin
  if b[i]=1 then begin
    c[1]:=9;
    c[2]:=8;
    c[3]:=7;
    c[4]:=4;
    c[5]:=3;
    c[6]:=1;
    c[7]:=0;
    k:=kl;
  end;
  else if b[i]=2 then begin
    c[1]:=8;
    c[2]:=2;
    k:=kl;
  end;
  else if b[i]=3 then begin
    c[1]:=9;
    c[2]:=8;
    c[3]:=3;
    k:=kl;
  end;
  else if b[i]=4 then begin
    c[1]:=9; c[2]:=8; c[3]:=4; k:=kl;
  end;
  else if b[i]=5 then begin
    c[1]:=9; c[2]:=8; c[3]:=5; k:=kl;
  end;
  else if b[i]=6 then begin
    c[1]:=8; c[2]:=6; k:=kl;
  end;
  else if b[i]=7 then begin
    c[1]:=9; c[2]:=8; c[3]:=7; c[4]:=0; k:=kl;
  end;
  else if b[i]=8 then begin
    c[1]:=8; k:=kl;
  end;
  else if b[i]=9 then begin
    c[1]:=9; c[2]:=8; k:=kl;
  end;
end;

```

Собираем в массив 'c' цифры числа наоборот

Узнаем количество цифр в числе

```

p:=0;
x:=1;
while
repeat
  if i<>1 then
    x:=x*10;
  inc(p);
until p<>kl-1;
j:=1;
while
while c[j]*x > M do
begin
  inc(j);
end;
M:=M - c[j]*10;
otv[kl]:=c[j];
inc(kl);
end;
for i:=1 to kl-1 do
  write(otv[i]);
end.

```

(12)

Собираем по цифрам число наоборот в массив c обратном

выводим

язык: Pascal

Задача №4

var a, nom1, nom2 : array[1..1000] of integer;

i, j, kol, u, v, l, d, x, zn : integer;

begin

read(n, m);

for i:=0 to n do

a[i]:=0;

kol:=0;

for i:=1 to m do

begin

read(zn);

if zn='P' then

begin

read(u, v);

for j:=u to v do

inc(~~to~~ a[j]);

inc(kol);

nom1[kol]:=u; // kol-номер элемента, k-1

nom2[kol]:=v;

end

else if zn='B' then

begin

read(x);

writeln(a[x]);

end

else

if zn='M' then

begin

read(l, d);

for j:=nom1[l] to nom2[l] do

dec(a[j]);

inc(a[d]);

end;

end;

end.

2