

**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



2  
696

1	2	3	4	Σ
3	10	φ	18	31

заполняется жюри!

62

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

заключительный этап

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

Город, в котором проводится Олимпиада Якутск

Дата 25.3.17

\*\*\*\*\*

**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

```

s = input().split()
a = []
if s[0] == '-' or s[0] == '+':
    for i in range(0, len(s), 2):
        if s[i] == '-':
            a.append(int(s[i+1]) - 1)
        else:
            a.append(int(s[i+1]))
else:
    a.append(int(s[0]))
    for i in range(1, len(s), 2):
        if s[i] == '-':
            a.append(int(s[i+1]) - 1)
        else:
            a.append(int(s[i+1]))
c = 0
for i in range(len(a)):
    c = c + a[i]
print(c)
    
```

3

### Задача №2.

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

Доступны следующие команды:

- НЦ<sub>x</sub>, КЦ – начало и конец цикла, x – количество выполнений цикла;
- НР, КР – начало рисования, конец рисования;
- ПВРТ\_ЛВ<sub>xx</sub>, ПВРТ\_ПР<sub>xx</sub> – поворот влево или вправо на xx градусов на месте;
- ВП<sub>xx</sub>, НЗ<sub>xx</sub> – вперед или назад на xx шагов (1 шаг = 1 см);
- ИСХ – команда возвращения робота в исходную позицию.
- ПРГ<sub>x</sub> – 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.

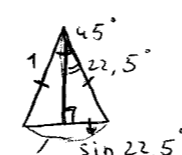
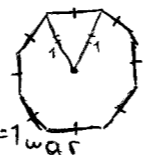
### Задача 2:

```

ПРГ Б1
  ВП 2 * sin 22,5°
  ПВРТ_ЛВ 135°
ПРГ Б2
  ПВРТ_ЛВ 45°
  ВП 4 * sin 22,5°
  ПВРТ_ЛВ 90°
ПРГ Б3
  ПВРТ_ЛВ 112,5°
  ВП 2
  ПВРТ_ЛВ 112,5°
  ВП 2 * sin 22,5°
НР
НЦ 8
  Б1
КЦ
НЦ 8
  Б2
КЦ
НЦ 2
  ПВРТ_ЛВ 22,5°
НЦ 4
  ВП 1/2
  ПВРТ_ЛВ 90°
КЦ
  ПВРТ_ПР 22,5°
  ВП 2 * sin 22,5°
КЦ
НЦ 4
  Б3
КЦ
  
```

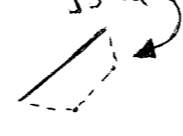
Комментарии:

программа Б1 рисует стороны 8-и угольника, где  
 $2 \cdot \sin 22,5^\circ$  – длина стороны, т.к. диагональ из центра = 1 шаг



Б2 – диагональ через 2 узла

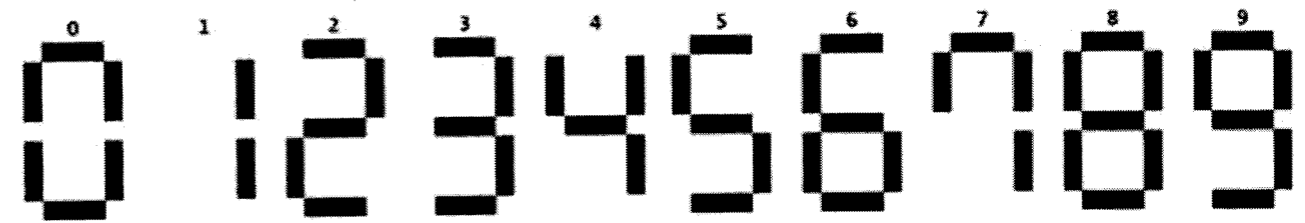
Б3 – диагональ через центр.



10

### 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

# Задача №4

## ПРОГРАММА

```

1 var n, m, x, y, i, j, k: integer; s: string;
2 begin
3   read(n, m); readln();
4   for i:=1 to n+1 do
5     a[i]:=0;
6     k:=0; j:=0;
7     while k < m do begin
8       read(s);
9       if s = 'P' then
10        begin
11          read(x); readln(y);
12          for i:=x to y+1 do
13            a[i]:=a[i]+1;
14            j:=j+1;
15            p[1, j]:=x; p[2, j]:=y;
16        end;
17       if s = 'M' then
18        begin
19          read(x); readln(y);
20          if y > 0 then begin
21          for i:=p[1, x] to p[1, x]+y do
22            a[i]:=a[i]-1; end;
23          p[1, x]:=p[1, x]+y; for i:=p[2, x] to p[2, x] do a[i]:=a[i]-1;
24          if y < 0 then begin
25            for i:=p[1, x] downto p[1, x]+y do
26              a[i]:=a[i]+1;
27            for i:=p[2, x] downto p[2, x]+y do
28              a[i]:=a[i]-1; end;
29          p[1, x]:=p[1, x]+y; p[2, x]:=p[2, x]+y;
30        end;
31       if s = 'B' then
32        begin
33          readln(x);
34          writeln(a[x]);
35        end;
36       k:=k+1;
37     end;
38 end.

```

## КОМЕНТАРИИ

- 1 Вводим переменные
- 3 Считываем n и m и уходим на след. строку.
- 4 Заполняем массив a, где 0 - не патрулируемый дом.
6. k - кумер для выхода из "Большого цикла", j - счетчик патрулей
7. начало "Большого цикла"
- 8 считываем букву (P/B/M), при этом не уходим на другую строку
- 9 проверяем, что за буква.
11. считываем цифры
- 12-13. т.к. у нас буква 'P', то мы помечаем дома, что их патрулируют ( $\neq 0$ )
15. массив p устроен так, что во всем первом ряду хранятся номера с какого нач. патруль, а во втором - на каком заканчивается
- 20-28 изменение патрульности дома с учетом положительного/отрицательного y (соответствует d из условия)
29. изменение патруля согласно ~~измене~~ диспетчеру
- 34 т.к. раньше мы не перезаписывали дома (a[i]), а прибавляли к ним y, то это значит, что a[i] показывает сколько на этом доме патрулей.
36. k:=k+1 значит, что мы считали строку полностью и что надо переходить на следующую.

