

**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 I d**  
Диспетчер дает команду группе с номером  $I$  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



1	2	3	4	Σ
6	5	7	20	38

76

заполняется жюри!

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

заключительный этап

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

Город, в котором проводится Олимпиада Санкт-Петербург

Дата 11.03.2017

\*\*\*\*\*

**Вариант 10**

\*\*\*\*\*

**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 одноптипные знаки (\*, /), но не менее 5 знаков.

**Входные данные:** Строка с математическим выражением.

**Выходные данные:** Число-ответ.

**Пример:**

**Ввод:** 2\*21/7\*8/4\*3

**Вывод:** 36

\*\*\*\*\*

**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

```

#include <string.h>
#include <math.h>
using namespace std;

int main() {
    string s;
    cin >> s;
    double res = 1;
    int i = 0;
    while (i < s.length()) {
        if (s[i] == '*') {
            double a = res;
            double b = s[i+1];
            res = a * b;
            i += 2;
        } else if (s[i] == '/') {
            double a = res;
            double b = s[i+1];
            res = a / b;
            i += 2;
        } else if (s[i] == '+') {
            double a = res;
            double b = s[i+1];
            res = a + b;
            i += 2;
        } else if (s[i] == '-') {
            double a = res;
            double b = s[i+1];
            res = a - b;
            i += 2;
        } else if (s[i] == '(') {
            double a = res;
            double b = s[i+1];
            res = a * b;
            i += 2;
        } else if (s[i] == ')') {
            double a = res;
            double b = s[i+1];
            res = a / b;
            i += 2;
        } else if (s[i] == ' ') {
            i++;
        } else {
            double a = res;
            double b = s[i];
            res = a * b;
            i++;
        }
    }
    cout << res;
    return 0;
}
    
```

6



Ujian Akhir

```
#include <iostream>
#include <vector>
using namespace std;
struct Point {int x, y;}
int main()
```

```
vector <Point> P;
int temp;
int x, y, n, m;
// read input
// read n
// read m
for (int i = 0; i < n; i++)
```

```
    cin >> x;
    if (ch == 'P') { // baca setiap baris P dan masukkan ke array P
        cin >> temp;
        P.push_back(temp);
```

```
    }
    if (ch == 'M') { // baca setiap baris M dan masukkan ke array M
        P[x-1][y] = x;
        P[y-1][x] = y;
```

```
    }
    if (ch == 'B') { // baca setiap baris B dan masukkan ke array B
        cin >> x;
        for (int i = 0; i < P.size(); i++)
            if (P[i][x] <= x && P[i][y] <= x) i++;
```

```
    }
    cout << endl;
}
```

20

```
int main()
return 0;
}
```



Zagrad 1/5, char C++

```

#include <string>
using namespace std;
char *strdup(char *str)
{
    char *ret = (char *) malloc(strlen(str) + 1);
    if (ret != NULL)
        strcpy(ret, str);
    return ret;
}
int main()

```

```

{
    char *x[100];
    int M, N;
    for (int i = 0; i < M; i++) // M, N, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99
    for (int i = 0; i < N; i++)
    int main()

```

```

{
    int n;
    n >= 0;
    for (int i = 0; i < strlen(x); i++)
    {
        cout << x[i];
        if (i % 10 < 9) cout << " ";
        if (i % 10 == 9) cout << "\n";
    }
    cout << "\n";
}

```

```

}
cout << x;
int main()
{
    cout << "086";
}

```

