

# Яндекс.Практикум, Алгоритмы: Спринт 1.

## Финальные задачи

🕒 6 авг 2020, 21:55:27  
старт: 5 авг 2020, 01:56:18

### А. Калькулятор

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Задание связано с обратной польской нотацией. Она используется для парсинга арифметических выражений. По сравнению с другим приемом, применяемым для данной задачи — использованием дерева операций, она является более компактной, так как в ней не используются скобки. Еще её иногда называют обратной польской записью или постфиксной нотацией. В постфиксной нотации операнды расположены перед знаками операций.

Пример 1:

3 4 +

будет равно 7, и означает  $3 + 4$

Пример 2:

10 2 4 \* -

будет равно 2, и означает  $10 - 2 * 4$

Разберем последний пример подробнее:

Знак \* стоит сразу после чисел 2 и 4, значит к ним нужно применить операцию, которую этот знак обозначает, то есть перемножить эти два числа. В результате получим 8

После этого выражение приобретет вид:

10 8 -

Операцию минус нужно применить к двум идущим перед ней числам, то есть 10 и 8. В итоге получаем 2.

Рассмотрим алгоритм более подробно. Для его реализации будем использовать стек.

Для вычисления значения выражения, записанного в обратной польской нотации, нужно считывать выражение слева направо и придерживаться следующих шагов:

1. Обработка входного символа: - Если на вход подан операнд, он помещается на вершину стека. - Если на вход подан знак операции, она выполняется над требуемым количеством значений из стека, взятых в порядке добавления. Результат выполненной операции помещается на вершину стека. 2. Если входной набор символов обработан не полностью, перейти к шагу 1. 3. После полной обработки входного набора символов результат вычисления выражения находится в вершине стека.

### Формат ввода

В единственной строке дано выражение, записанное в обратной польской нотации. Числа и арифметические операции отделены друг от друга пробелами.

На вход могут подаваться операции: +, -, \*, / и числа, по модулю не превосходящие 10000.

Гарантируется, что значение промежуточных выражений в тестовых данных по модулю не больше 50000.

### Формат вывода

Единственное число - значение выражения.

#### Пример 1

**Ввод**

2 1 + 3 \*

**Вывод**

9

## Пример 2

Ввод

7 2 + 4 \* 2 +

Вывод

38

## Примечания

Операция деления целочисленная. То есть, например, 12 5 / будет 2.

Решение должно быть реализовано с использованием структуры данных стек.

Язык GNU c++17.3

Набрать здесь

Отправить файл

```
1 #include <iostream>
2 #include <string>
3 #include <sstream>
4 #include <set>
5
6 using namespace std;
7
8 /**
9  * @brief структура элемента связанного списка
10  */
11 struct Node
12 {
13     Node( int value ) : mValue( value ) {}
14     Node( int value, Node* next ) : mValue( value ), mNext( next ) {}
15
16     int mValue; // значение элемента
17     Node* mNext; // указатель на следующий элемент
18 };
19
20 /**
21  * @brief структура стека
22  */
23 struct Stack
24 {
25     Stack() : mTop( nullptr ), mSize( 0 ) {}
26
27     /**
28      * @brief определить, пустой ли стек
29      * @return true / false
30      */
31     bool Empty()
32     {
33         return mSize == 0 ? true : false;
34     }
35
36     /**
37      * @brief положить в стек новый элемент
38      * @param value - значение элемента
39      */
40 }
```

Отправить

Следующая