Домашнее задание №1 по дисциплине: Разработка приложений на языке C#

**Задание:**

Нужно реализовать консольный калькулятор, умеющий вычислять выражение, подаваемое на STDIN.

Требуется реализовать:

* сложение;
* вычитание;
* умножение;
* деление;
* поддержка скобок.

Нужно написать тесты, которые покрывают все операции.

**Пример:**

**"(2+3)-4" => 1**

**"4-(2\*3)" => 2**

**Варианты:**

Четные по списку - методом Бауэра-Земельзона

Нечетные по списку - стековым методом

**Отчет:**

1. Введение.
2. Описание работы программы.
3. Листинг программы.
4. Демонстрация работы приложения.
5. Демонстрация пройдённых тестов.
6. Вывод.

**Теоретическая часть**

**Стековый метод:**

Пусть задана некоторая произвольная строка:

....αβ....

Если два символа строки α, β ∈*V* расположены рядом в сентенциальной форме, то между ними возможны следующие отношения, названные отношениями предшествования:

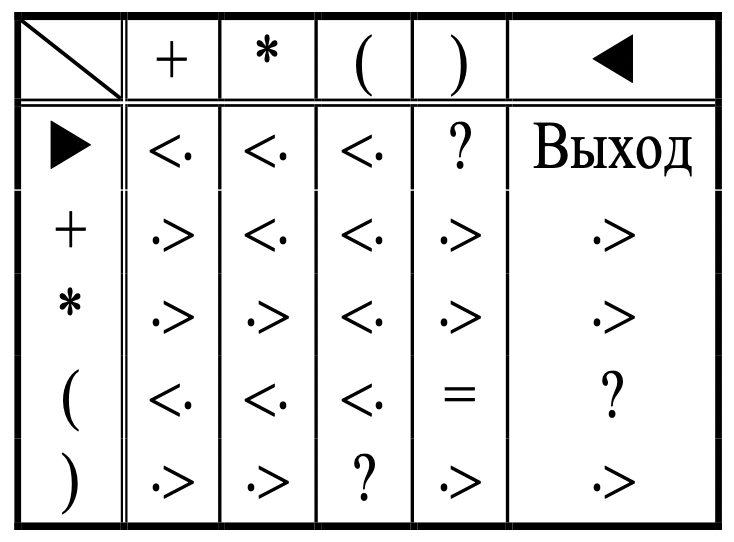
1. α принадлежит основе, а β – нет, т. е. α – конец основы: α ⋅> β;
2. β принадлежит основе, а α – нет, т. е. β – начало основы: α <⋅ β ;
3. α и β принадлежит одной основе, т. е. α =⋅ β;
4. α и β не могут находиться рядом в сентенциальной форме (ошибка).

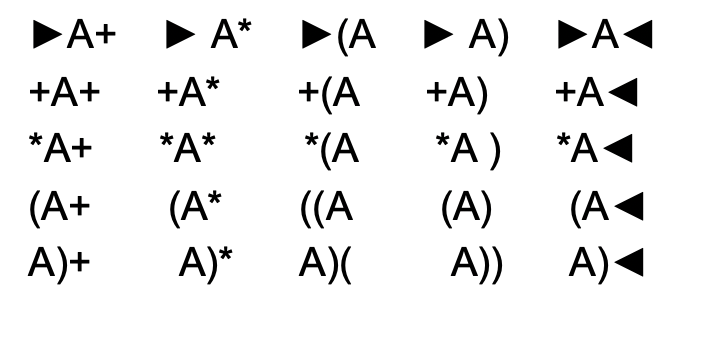
<Выражение> ::= <Терм>|<Выражение>+<Терм>|<Выражение>- <Терм>

<Терм> ::= <Множитель> | <Терм>\* <Множитель> | <Терм> / <Множитель>

<Множитель> ::= (<Выражение>) | <Идентификатор>

<⋅ - начало основы; ⋅> - конец основы; = - одна основа; ? – ошибка

****

****

**Пример разбора выражения стековым методом ►d+c\*(a+b)◄**



**Польская запись. Алгоритм Бауэра-Замельзона:**

Польская запись представляет собой последовательность команд двух типов:

КI, где I – идентификатор операнда – выбрать число по имени I и заслать его в стек операндов;

Kξ, где ξ – операция – выбрать два верхних числа из стека операндов, произвести над ними операцию ξ и занести результат в стек операндов.

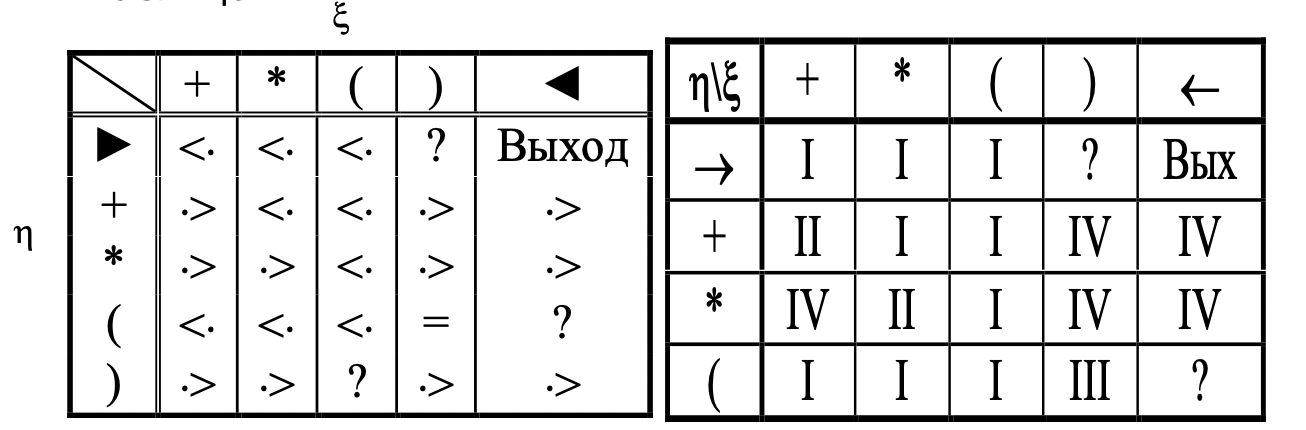
Пример:

A+B\*C ⇒ KAKBKCK\*K+

**Построение польской записи:**

а) если символ – операнд, то вырабатывается команда КI,

б) если символ – операция, то выполняются действия согласно таблице:



Операции:

I – заслать ξ в стек операций и читать следующий символ;

II – генерировать Kη , заслать ξ в стек операций и читать следующий символ;

III – удалить верхний символ из стека операций и читать следующий символ;

IV – генерировать Kη и повторить с тем же входным символом.

Построение троек для **(a+b\*c)/d.**



**Ka Kb Kc K\* K+ Kd K/** или **abc\*+d/**

Выполнение операций польской записи:



**Пример написания тестов:**

using System;

using System.IO;

namespace DZ\_1

{

class Program {

static void Main(string[] args)

{

string[] inputs = {"1+2", "(234-11)\*34", "6\*6/6"}; // Пример

string[] outputs = {"3", "7582", "6"}; // Пример

for(int i = 0; i < inputs.Length; i++) {

Console.WriteLine(inputs[i] + "=" + outputs[i]);

if (algorithm(inputs[i]) == outputs[i]) {

Console.WriteLine("OK");

} else {

Console.WriteLine("Faild");

}

}

}

}

}

Результат округлять до целого значения!