**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И ТЕХНОЛОГИЙ**

**ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ**

**Контрольная работа №2**

**Вариант**

**по предмету «Алгоритмы и структуры данных»**

**Выполнил: студент гр. 5130904/30002 Севостьянова А.В.**

**Руководитель Череповский Д.К.**

**Санкт-Петербург**

**2024 г.**

1. **Общая постановка задачи**

Реализуйте функцию, указанную в варианте

В реализации используйте шаблон «ограниченный стек» (см. Задание 2)

Выполните тестирование

# **Функция перевода арифметического выражения из инфиксной формы в постфиксную (обратную польскую запись).**

Функция должна формировать выражение в постфиксной форме и генерировать исключение, если возникает ошибка (например, при переполнении стека, неправильной последовательности операндов и операций, …)

Считать, что выражение состоит из операндов (***одноразрядных*** десятичных чисел), знаков ***бинарных*** операций ( +, -, \*, / ) и ***круглых*** скобок.

Перевод выполнить за ***один проход*** по выражению, в процессе выполнения которого проверить правильность операндов, допустимость операций, правильность скобок.

Прототип функции:

void getPostfixFromInfix (const char\* infix, char\* postfix,

const size\_t stackSize);

или

void getPostfixFromInfix (const string& infix, string& postfix,

const size\_t stackSize);

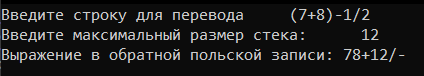
infix - выражение в инфиксной форме

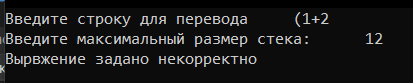
postfix - выражение в постфиксной форме

stackSize - максимальный размер стека операций

# **Алгоритм получения обратной польской записи *(!!! Корректное выражение)***

1. Подготовить пустой стек.
2. Просматривать исходную строку символов слева направо.
3. Операнды переписывать в выходную строку, а знаки бинарных операций заносить в стек по следующему правилу:
4. если стек пуст, то операцию из входной строки занести в стек;
5. если стек не пуст, операцию занести в стек, предварительно вытолкнуть из стека в выходную строку все операции с большим или равным приоритетом;
6. если очередной символ из исходной строки - открывающая скобка, то занести его в стек;
7. если очередной символ из исходной строки – закрывающая скобка, то вытолкнуть из стека все операции до ближайшей открывающей скобки, сами скобки в выходную строку не переписывать.
8. В завершении выражения оставшиеся операции из стека переносятся в выходную строку.
9. **Работа программы**

****

****