

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: стек и очередь**

Студентка гр. 8381

Ивлева О.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

## Цель работы.

Ознакомиться с методами перевода выражения в префиксную форму.

## Задание.

Рассматривается выражение следующего вида:

$\langle \text{выражение} \rangle ::= \langle \text{терм} \rangle \mid \langle \text{терм} \rangle + \langle \text{выражение} \rangle \mid$

$\langle \text{терм} \rangle \square \langle \text{выражение} \rangle$

$\langle \text{терм} \rangle ::= \langle \text{множитель} \rangle \mid \langle \text{множитель} \rangle * \langle \text{терм} \rangle$

$\langle \text{множитель} \rangle ::= \langle \text{число} \rangle \mid \langle \text{переменная} \rangle \mid ( \langle \text{выражение} \rangle ) \mid$

$\langle \text{множитель} \rangle \wedge \langle \text{число} \rangle$

$\langle \text{число} \rangle ::= \langle \text{цифра} \rangle$

$\langle \text{переменная} \rangle ::= \langle \text{буква} \rangle$

Такая форма записи выражения называется инфиксной.

Постфиксной (префиксной) формой записи выражения  $aDb$  называется запись, в которой знак операции размещен за (перед) операндами:  $abD$  ( $Dab$ ).

## Примеры

Инфиксная	Постфиксная	Префиксная
$A+b$	$ab+$	$+ab$
$a*b+c$	$ab*c+$	$+*abc$
$a*(b+c)$	$abc+*$	$*a+bc$
$a+b^c^d*e$	$abc^d^e*+$	$+a*^b^cde.$

Отметим, что постфиксная и префиксная формы записи выражений не содержат скобок.

## Требуется:

в) перевести выражение, записанное в обычной (инфиксной) форме в заданном текстовом файле `infix`, в постфиксную форму и в таком виде записать его в текстовый файл `postfix`;

## **Основные теоретические положения.**

Стек — это структура данных, которая работает по принципу FILO (first in — last out; первый пришел — последний ушел).

В стеке элемент, который вошел самый первый — выйдет самым последним. Получается, если вы добавили три элемента в стек первым будет удален последний добавленный элемент.

## **Выполнение работы.**

Написание работы производилось на базе операционной системы Windows 10 в среде разработки QtCreator с использованием фреймворка Qt.

Для реализации программы был разработан графический интерфейс с помощью встроенного в QtCreator UI-редактора. Была добавлена кнопка, при нажатии на которую, происходит преобразование инфиксной записи выражения в постфиксную.

Была написана функция postfix(), которая получает на вход исходную строку и флаг, нужно ли выводить на экран дополнительную информацию.

### **Алгоритм**

Пока есть ещё символы для чтения:

- Читаем очередной символ.
- Если символ является числом, добавляем его к выходной строке..
- Если символ является открывающей скобкой, помещаем его в стек.
- Если символ является закрывающей скобкой:

До тех пор, пока верхним элементом стека не станет открывающая скобка, выталкиваем элементы из стека в выходную строку. При этом открывающая скобка удаляется из стека, но в выходную строку не добавляется. Если стек закончился раньше, чем мы встретили открывающую скобку, это означает, что в выражении либо неверно поставлен разделитель, либо не согласованы скобки.

Когда входная строка закончилась, выталкиваем все символы из стека в выходную строку. В стеке должны были остаться только символы операторов; если это не так, значит в выражении не согласованы скобки.

### **Оценка эффективности алгоритма.**

Алгоритм, реализованный в программе, имеет линейную зависимость от длины строки, то есть сложность оценивается как  $O(n)$ , так как алгоритм проходит по строке только один раз.

### **Тестирование приложения.**

Вид программы после запуска:

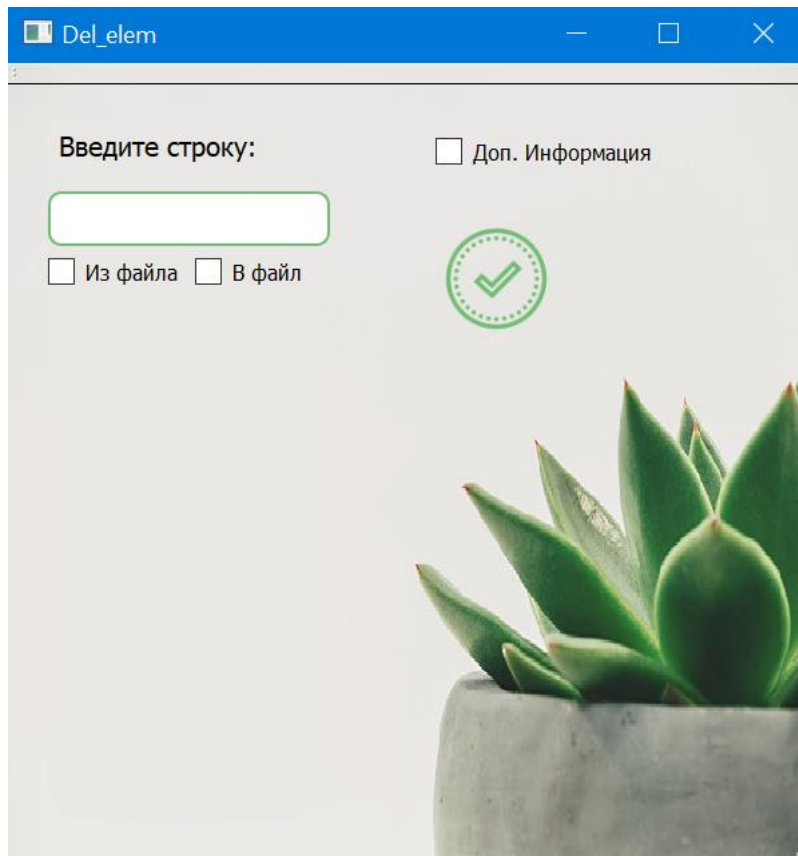


Таблица 1 – Тестирование программы

Входная строка	Вывод
3+4	34+ Stack: +    Stack:
(5-4)+2	54-2+ Stack: (    Stack: (-    Stack: (    Stack:    Stack:    + Stack:
2*(3+4)*5	234+.5. Stack: .    Stack: .(    Stack: .(+    Stack: .(    Stack: Stack:    Stack: .    Stack:
a+b^c^d*e	abc^d^e.+ Stack: +    Stack: +^    Stack: +    Stack: +^    Stack: +Stack: +.    Stack: +    Stack:

### Вывод.

В ходе выполнения лабораторной работы был изучен алгоритм преобразования инфиксного выражения в постфиксное на основе стека.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.c

```
#include "mainwindow.h"
#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

Название файла: mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include "functions.h"

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_Hello_clicked();

private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H
```

Название файла: functions.h

```
#include <string>
#include <iostream>
using namespace std;

string postfix(string, bool);
Название файла: functions.cpp
#include "functions.h"

string postfix(string inf, bool info){
```

```

int i = 0;
cout << info << endl;
string stack, str, str2;
stack = str = str2 = "";
while (inf[i] != '\0') {
    if (isalnum(inf[i]))
        str += inf[i];
    if (inf[i] == '+' || inf[i] == '-' || inf[i] == '*' || inf[i]
== '^') {
        if (inf[i] == '*') { //Замена умножения
            inf[i] = '.';
        }
        if (stack.empty()){
            stack += inf[i];
            str2 += "Stack: ";
            str2 += stack;
            str2 += "\t";
        }
        else {
            while (stack.back() >= inf[i]) { //выталкиваем все
элементы из стэка
//До тех пор, пока приоритет верхнего элемента
не будет
            str += stack.back();
//ниже входящего символа
            stack.pop_back();
            str2 += "Stack: ";
            str2 += stack;
            str2 += "\t";
        }
            stack += inf[i];
            str2 += "Stack: ";
            str2 += stack;
            str2 += "\t";
        }
    }
    if (inf[i] == '(') { //Если символ - открывающая скобка
        stack += inf[i]; //добавляем ее в стэк
        str2 += "Stack: ";
        str2 += stack;
        str2 += "\t";
    }
    if (inf[i] == ')') { //Если встречающийся символ -
закрывающая скобка
        while (stack.back() != '(') { // пока в стеке не
встретится открывающая скобка
            // выталкиваем в главную строку все символы
            str += stack.back();
            stack.pop_back();
            str2 += "Stack: ";
            str2 += stack;
            str2 += "\t";
        }
        stack.pop_back(); //Удаляем открывающую скобку из стека
        str2 += "Stack: ";
        str2 += stack;
    }
}

```

```

        str2 += "\t";
    }
    i++;
}
while (!stack.empty()) { //При окончании строки выталкиваем все
СИМВОЛЫ ИЗ СТЕКА
    str += stack.back();
    stack.pop_back();
    str2 += "Stack: ";
    str2 += stack;
    str2 += "\t";
}
if(info){
    str += "\n";
    str += str2;
}
return str;
}

```

### Название файла: mainwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"

```

```

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ui->output->setWordWrap(true);
    QPixmap bkgnd("D:\\prog\\Styles\\plant-2004483_640.jpg");
    bkgnd = bkgnd.scaled(this->size(), Qt::IgnoreAspectRatio);
    QPalette palette;
    palette.setBrush(QPalette::Background, bkgnd);
    this->setPalette(palette);
}

```

```

MainWindow::~MainWindow()
{
    delete ui;
}

```

```

void MainWindow::on_Hello_clicked()
{
    string str;
    if (!ui->checkBox_2->isChecked()){
        str = qPrintable(ui->input->text());
    }
    else {
        ifstream fin; // создаем объект класса ifstream (считать)
        fin.open("D:\\prog\\cpp\\lab1\\text.txt"); // открываем файл
        для считывания
        fin >> str;
        fin.close(); // закрываем файл
    }
    string output = "";
    output += postfix(str, ui->checkBox->isChecked());
}

```



```

        output += "\n";
        output += "\n";
        if (ui->checkBox_3->isChecked()){
            ofstream fin2; //дозаписать
            fin2.open("D:\\prog\\cpp\\lab1\\text.txt", ios::app); //
открываем файл и дозываетываем в него
            fin2 << endl;
            fin2 << output;
            fin2.close(); // закрываем файл
        }
        ui->output->setText(QString::fromStdString(output));
    }
}

```

### Название файла: mainwindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>MainWindow</class>
    <widget class="QMainWindow" name="MainWindow">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>600</width>
                <height>600</height>
            </rect>
        </property>
        <property name="focusPolicy">
            <enum>Qt::NoFocus</enum>
        </property>
        <property name="windowTitle">
            <string>Del_elem</string>
        </property>
        <property name="autoFillBackground">
            <bool>false</bool>
        </property>
        <property name="styleSheet">
            <string notr="true"/>
        </property>
        <widget class="QWidget" name="centralWidget">
            <widget class="QPushButton" name="Hello">
                <property name="geometry">
                    <rect>
                        <x>320</x>
                        <y>100</y>
                        <width>91</width>
                        <height>91</height>
                    </rect>
                </property>
                <property name="cursor">
                    <cursorShape>PointingHandCursor</cursorShape>
                </property>
                <property name="styleSheet">
                    <string notr="true">border-image:
url(:/C:/Users/Олеся/Downloads/icons8-ok-100_1.png);</string>
                </property>
                <property name="text">

```

```

    <string/>
  </property>
  <property name="autoRepeat">
    <bool>false</bool>
  </property>
</widget>
<widget class="QLabel" name="output">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>250</y>
      <width>201</width>
      <height>301</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">font: 6pt &quot;MS Shell Dlg 2&quot;;
border-radius: 30%;
background-color: #e9e8e4;
</string>
  </property>
  <property name="text">
    <string/>
  </property>
</widget>
<widget class="QLineEdit" name="input">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>80</y>
      <width>211</width>
      <height>41</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">border-radius: 10%;
border: 2px solid #72be74;</string>
  </property>
</widget>
<widget class="QLineEdit" name="inputChar">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>190</y>
      <width>211</width>
      <height>41</height>
    </rect>
  </property>
  <property name="styleSheet">
    <string notr="true">border-radius: 10%;
border: 2px solid #72be74;</string>
  </property>
</widget>
<widget class="QCheckBox" name="checkBox">
  <property name="geometry">
    <rect>

```

```

        <x>320</x>
        <y>40</y>
        <width>191</width>
        <height>21</height>
    </rect>
</property>
<property name="cursor">
    <cursorShape>PointingHandCursor</cursorShape>
</property>
<property name="text">
    <string>Доп. Информация</string>
</property>
</widget>
<widget class="QLineEdit" name="lineEdit">
    <property name="geometry">
        <rect>
            <x>30</x>
            <y>30</y>
            <width>181</width>
            <height>31</height>
        </rect>
    </property>
    <property name="styleSheet">
        <string notr="true">
font: 10pt "MS Shell Dlg 2";
border-radius: 10%;
background-color: #e9e8e4;
</string>
        </property>
        <property name="text">
            <string>Введите строку:</string>
        </property>
    </widget>
    <widget class="QLineEdit" name="lineEdit_2">
        <property name="geometry">
            <rect>
                <x>30</x>
                <y>140</y>
                <width>181</width>
                <height>31</height>
            </rect>
        </property>
        <property name="styleSheet">
            <string notr="true">
font: 10pt "MS Shell Dlg 2";
border-radius: 10%;
background-color: #e9e8e4;</string>
            </property>
            <property name="text">
                <string>Введите символ:</string>
            </property>
        </widget>
    </widget>
    <widget class="QMenuBar" name="menuBar">
        <property name="geometry">
            <rect>

```

```

        <x>0</x>
        <y>0</y>
        <width>600</width>
        <height>17</height>
    </rect>
</property>
</widget>
<widget class="QToolBar" name="mainToolBar">
    <attribute name="toolBarArea">
        <enum>TopToolBarArea</enum>
    </attribute>
    <attribute name="toolBarBreak">
        <bool>>false</bool>
    </attribute>
</widget>
<widget class="QStatusBar" name="statusBar"/>
</widget>
<layoutdefault spacing="6" margin="11"/>
<resources/>
<connections/>
</ui>

```