Федеральное государственное автономное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ТЕЛЕКОММУНИКАЦИИ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Ассистент ИКТИБ Студент группы КТбо1-2

Кочубей Д. С. Воробьев В. М.

«26» Октября 2023 г. «26» Октября 2023 г.

**Индивидуальная работа**

На тему:

«АЛГОРИТМЫ СОРТИРОВКИ»

Вариант 8

**Таганрог 2023**

# Содержание

Содержание 2

Техническое задание 3

Цель задания 3

Задача 3

Ход работы 4

Алгоритм 4

# Техническое задание

**Цель**

Целью данной лабораторной работы является выработка у студента практических навыков работы со стеком на примере построения обратной польской записи арифметических и логических выражений.

## Задача

Программа должна реализовать перевод в ОПЗ простых арифметических и логических выражений.

# Ход работы

## Алгоритм

Создаётся стек и 3 дополнительных программы к нему: помещение и изымание элемента из стэка, а также программа, которая возвращает приоритет внесённого в неё операнда. Далее рассматривается каждый элемент внесённой строки по следующему принципу:

1.1. Если символ операнд, то он перемещается в выходную строку.

1.2. Если символ операция и стек пуст, то он проталкивается в стек.

1.3. Если символ операция и стек не пуст, а приоритет символа операции меньше, чем у символа операции вершины стека, то прочитанный символ проталкивается и стек.

1.4. Если символ операция и стек не пуст, а приоритет символа вершины стека, меньше приоритета входного символа, то из стека в выходную строку выталкиваются все символы операций с меньшим или равным приоритету входного символа, после чего входной символ операции заносится в стек.

1.5. Если входной символ левая открывающая скобка, то она проталкивается в стек.

1.6. Если входной символ правая закрывающая скобка, то она выталкивает из стека в выходную строку все символы до левой открывающей скобки. сами скобки уничтожаются и в выходную строку не попадают. 15

1.7. Если входной символ - маркер конца строки, то из стека в выходную строку выталкиваются все символы.

2. Результат выходной строки. Конец алгоритма.

## Блок-схема

Ниже представлены блок-схемы основной и дополнительных программ (Рисунок 1-2).

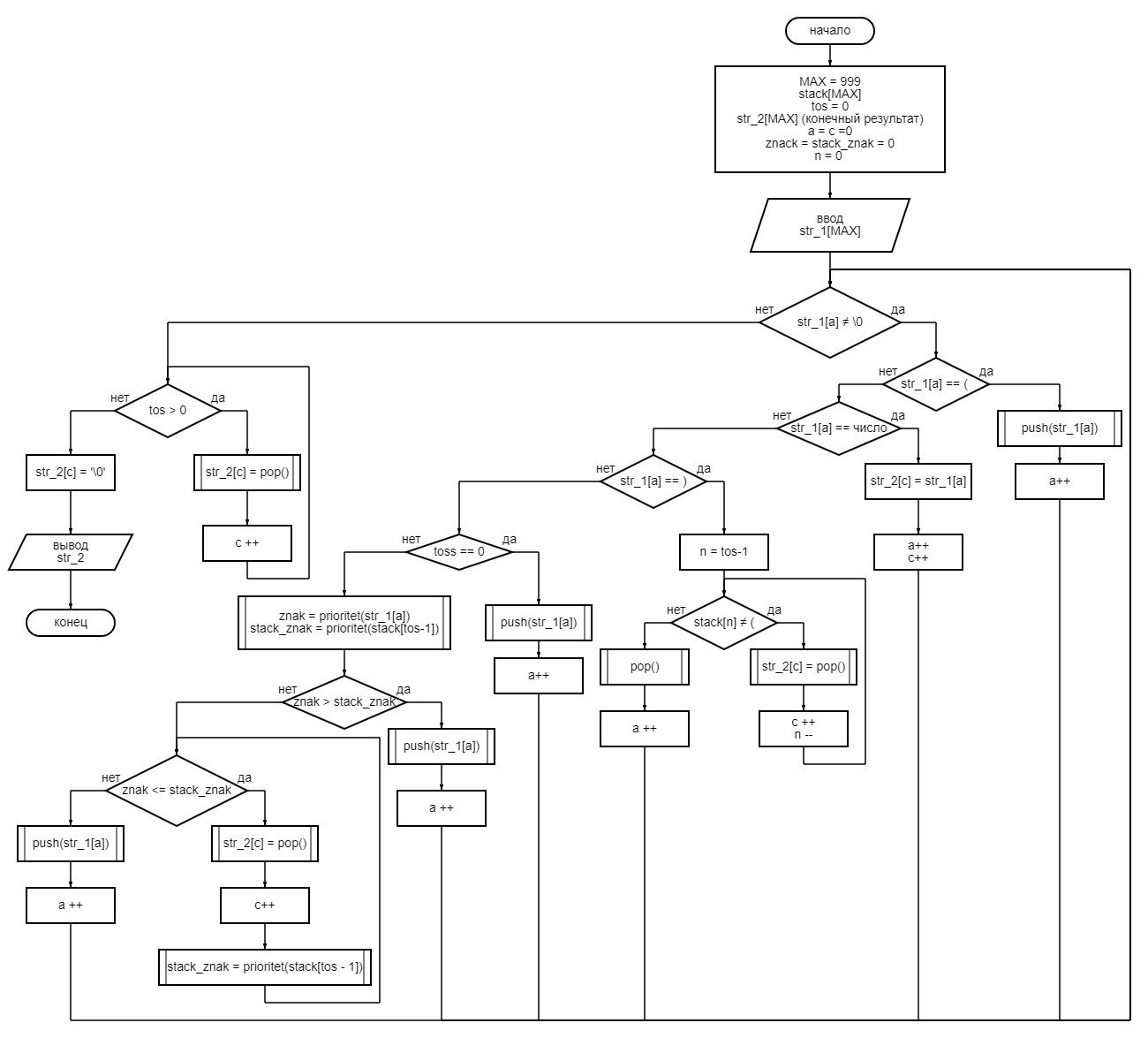


Рисунок 1 – Блок-схема основной программы

## 

Рисунок 2 – Блок-схема дополнительных программ

## Написание программы

Создаем стек и 3 дополнительные программы для работы с ним (push, pop, prioritet). Далее создаётся цикл, который рассмотрит каждый элемент входной строки (элементы будут перебираться, пока значение не будет равно “\0”, также тут используются счётчики “a” и “c”).

С помощью функции “isdiget” будет проверяться, является ли элемент числом. Если является, то элемент записывается в выходную строку. Если нет, что проверяется, является ли элемент символом “(”. Если да, то он помещается в стек (с помощью функции push), если нет, то проверяется, является ли элемент “)”. Если является, то с помощью цикла while мы перебираем все элементы стека до элемента “)” и отправляем их в выходную строку (элемент “)” уничтожается).

Если элемент является другой операцией, то мы проверяем количество элементов в стеке с помощью переменной “tos”. Если она равна 0 (стек пуст), то элемент помещается в стек. Если нет, то пользуясь функцией “prioritet”, а также переменными “znak” (приоритет рассматриваемого элемента) и “stack\_znak” (приоритет элемента вершины стека) мы либо помещаем элемент в стек (в случае, если znak > stack\_znak), либо выносим в выходную строку все элементы с меньшим или равным приоритетом (znak <= stack\_znak).

После прохождения всей строки, если в стеке ещё есть элементы, то они тоже помещаются в выходную строку. Строка выводится на экрав. Конец программы.

## Пример работы программы

Пример работы программы представлен ниже (Рисунок 3).

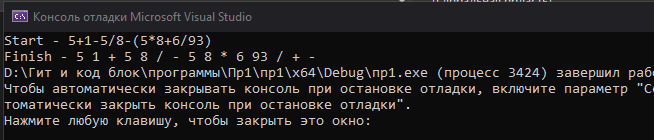


Рисунок 3 - Пример работы программы

# Вывод

По итогу я смог выработать у себя практические навыки работы со стеком на примере построения обратной польской записи арифметических и логических выражений, а также разработал программу перевода арифметической записи число в ОПЗ.

# Приложение

Текст программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <ctype.h>

#include <string.h>

int tos = 0, MAX = 999;

char stack[MAX], w[2];

void push(char i)

{

if (tos >= MAX) {

printf("Stack full\n");

return;

}

stack[tos] = i;

tos++;

}

char pop(void)

{

tos--;

if (tos < 0)

{

printf("Stack empty\n");

return 0;

}

w[0] = stack[tos];

stack[tos] = '\0';

return w[0];

}

int prioritet(char f)

{

if (f == '(')

{

return 0;

}

if (f == ')')

{

return 1;

}

if (f == '+' or f == '-')

{

return 2;

}

if (f == '\*' or f == '/')

{

return 3;

}

if (f == '^')

{

return 4;

}

}

int main()

{

char str\_1[MAX];

char str\_2[MAX];

int a = 0, c = 0, znak = 0, stack\_znak = 0, n;

strcpy(str\_1, "5+1-5/8-(5\*8+6/93)");

while (str\_1[a] != '\0')

{

if (str\_1[a] == '(')

{

push(str\_1[a]);

a++;

continue;

}

if (isdigit(str\_1[a]))

{

str\_2[c] = str\_1[a];

a++;

c++;

if (not(isdigit(str\_1[a])) and str\_1[a] != '\0')

{

str\_2[c] = ' ';

c++;

}

continue;

}

if (str\_1[a] == ')')

{

n = tos-1;

while (stack[n] != '(')

{

str\_2[c] = pop();

str\_2[c + 1] = ' ';

c += 2;

n--;

}

pop();

a++;

continue;

}

if (tos == 0)

{

push(str\_1[a]);

a++;

continue;

}

znak = prioritet(str\_1[a]);

stack\_znak = prioritet(stack[tos-1]);

if (znak > stack\_znak)

{

push(str\_1[a]);

a++;

continue;

}

else

{

for (int m = tos; m > 0; m--)

{

if (znak <= stack\_znak)

{

str\_2[c] = pop();

str\_2[c + 1] = ' ';

c += 2;

stack\_znak = prioritet(stack[tos - 1]);

}

else

{

break;

}

}

push(str\_1[a]);

a++;

continue;

}

}

while (tos > 0)

{

str\_2[c] = pop();

str\_2[c + 1] = ' ';

c += 2;

}

str\_2[c] = '\0';

printf("Start - %s\n", str\_1);

printf("Finish - %s", str\_2);

return 0;

}