Федеральное государственное автономное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ТЕЛЕКОММУНИКАЦИИ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Старший преподаватель кафедры САиТ Студент группы КТбо1-2

Лапшин В.С. Кукса К.С.

«\_\_» \_\_\_\_\_\_\_\_\_\_2023 г. «\_\_» \_\_\_\_\_\_\_\_\_\_2023 г.

**ЛАБОРАТОРНАЯ РАБОТА №2**

По дисциплине

«Основы алгоритмизации и программирования»

Вариант 4

На тему

«Работа со стеком»

**Таганрог 2023**

**Цель**: выработка практических навыков работы со стеком на примере построения обратной польской записи арифметических и логических выражений.

**Задачи**: написать программу, реализующую перевод в ОПЗ простых арифметических и логических выражений. Ввод и вывод данных организовать во внешние файлы. Запрограммировать работу стека на основе двух функций push () и pop (), программно реализовать алгоритм Дейкстры..

**Ход работы**:

**Составление математической модели**:

Алгоритм перевода в ОПЗ:

1. Читать очередной символ входной строки, пока не будет достигнут конец выражения:

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.

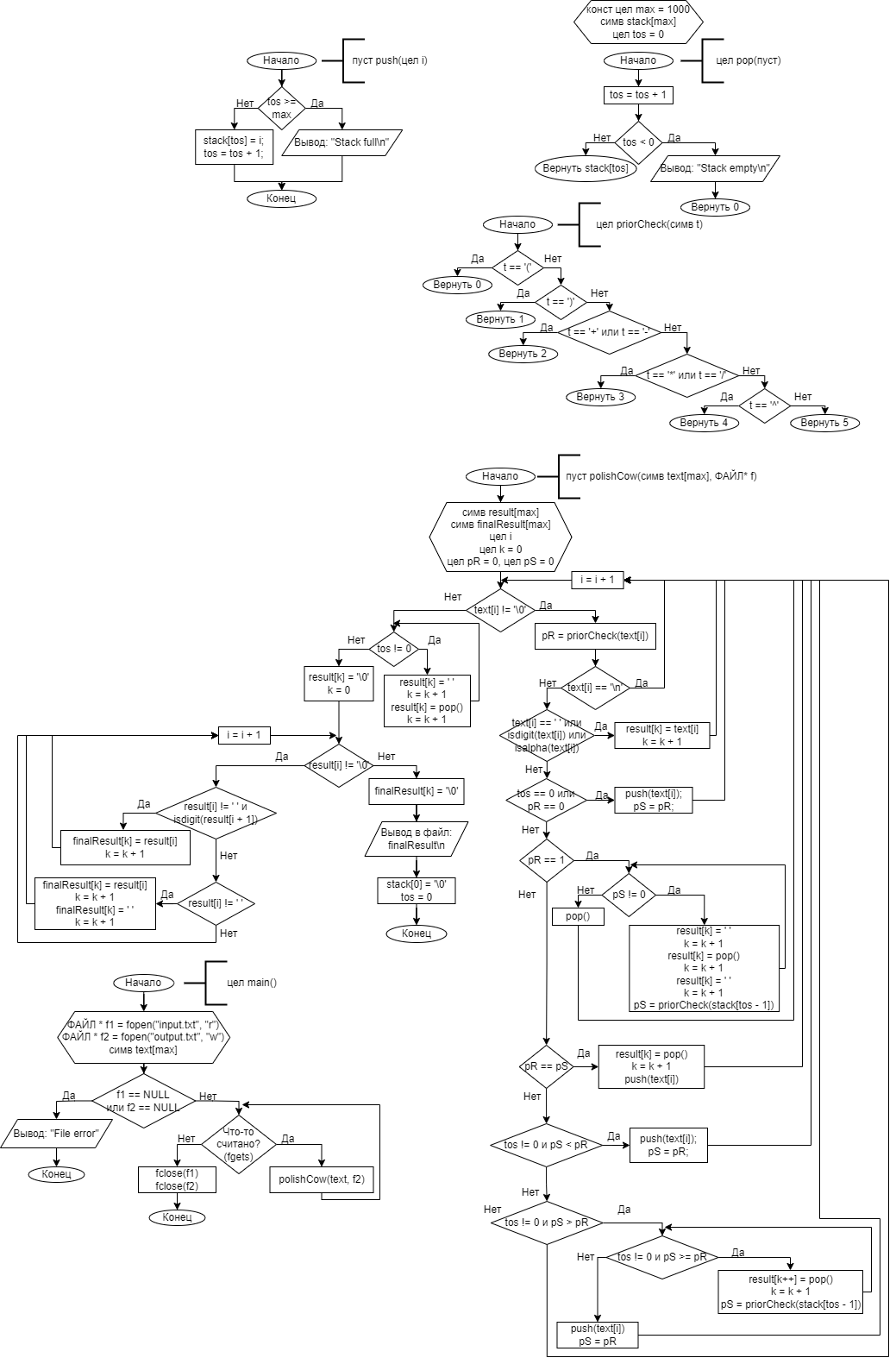
1.5. Если входной символ левая открывающая скобка, то она проталкивается в стек.

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.

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

2. Результат выходной строки. Конец алгоритма.

**Проектирование.**

****

**Реализация.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <ctype.h>

#include <string.h>

const int max = 1000;

char stack[max];

int tos = 0;

void push(int i)

{

if (tos >= max) {

printf ("Stack full\n"); return;

}

stack[tos] = i;

tos++;

}

int pop(void)

{

tos--;

if (tos < 0) {

printf("Stack empty\n"); return 0;

}

return stack[tos];

}

int priorCheck(char t)

{

if (t == '(') return 0;

else if (t == ')') return 1;

else if (t == '+' || t == '-') return 2;

else if (t == '\*' || t == '/') return 3;

else if (t == '^') return 4;

else return 5;

}

void polishCow(char text[max], FILE\* f)

{

char result[max], finalResult[max];

int i, k = 0, pR = 0, pS = 0;

for (i = 0; text[i] != '\0'; i++) {

pR = priorCheck(text[i]);

if (text[i] == '\n') {}

else if (text[i] == ' ' || isdigit(text[i]) || isalpha(text[i])) {

result[k++] = text[i];

}

else if (tos == 0 || pR == 0) {

push(text[i]);

pS = pR;

}

else if (pR == 1) {

while (pS != 0) {

result[k++] = ' ';

result[k++] = pop();

result[k++] = ' ';

pS = priorCheck(stack[tos - 1]);

}

pop();

}

else if (pR == pS) {

result[k++] = pop();

push(text[i]);

}

else if (tos != 0 && pS < pR) {

push(text[i]);

pS = pR;

}

else if (tos != 0 && pS > pR) {

while (tos != 0 && pS >= pR) {

result[k++] = pop();

pS = priorCheck(stack[tos - 1]);

}

push(text[i]);

pS = pR;

}

}

while (tos != 0) {

result[k++] = ' ';

result[k++] = pop();

}

result[k] = '\0';

k = 0;

for (i = 0; result[i] != '\0'; i++)

if (result[i] != ' ' && isdigit(result[i + 1])) {

finalResult[k++] = result[i];

}

else if (result[i] != ' ') {

finalResult[k++] = result[i];

finalResult[k++] = ' ';

}

finalResult[k] = '\0';

fprintf(f, "%s\n", finalResult);

stack[0] = '\0';

tos = 0;

return;

}

int main()

{

FILE \* f1 = fopen("input.txt", "r"), \* f2 = fopen("output.txt", "w");

char text[max];

if (f1 == NULL || f2 == NULL) {

printf("File error");

return 0;

}

while (fgets(text, max, f1))

polishCow(text, f2);

fclose(f1);

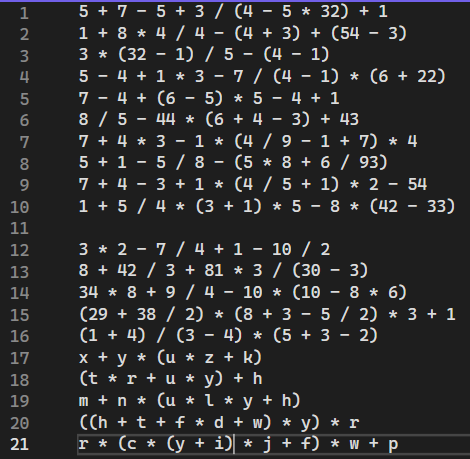
fclose(f2);

return 0;

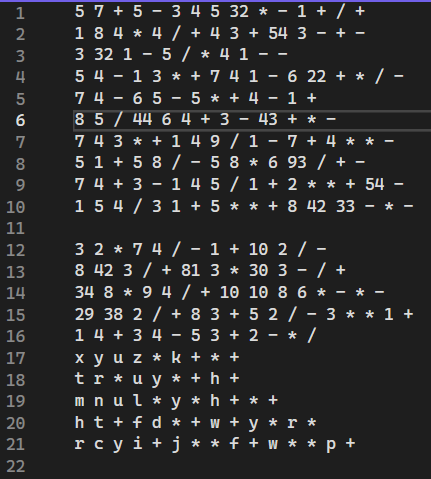
}

**Результат работы программы.**

Входные данные:



Выходные данные:



**Вывод:**

В ходе работы была создана программа, переводящая входящие арифметические и логические выражения в ОПЗ. Был программно реализован алгоритм Дейкстры. Программа была протестирована на арифметических и логических выражениях, вводимых из файла. Программа успешно перевела большую часть выражений, однако имеет в результате своей работы неточности:

1. некорректно воспринимаются действия в скобках, перед которой стоит операция вычитания;
2. иногда некорректно воспринимаются границы деления (действия после деления в некоторых случаях могут оказаться внутри делителя).

Причины проблем локализовать не удалось.

В остальных случаях программа функционирует верно и свою задачу выполняет.