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.**

Изображение выглядит как диаграмма

Автоматически созданное описание

**Реализация.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <ctype.h>

#include <string.h>

#include <stdio.h>

const int max = 1000;

char stack[max];

int tos = 0;

void push(int i)

{

if (tos >= max)

printf("Stack full\n");

else

stack[tos++] = i;

}

char pop(void)

{

tos--;

if (tos < 0)

printf("Stack empty\n");

else

return stack[tos];

return 0;

}

int priorCheck(char t)

{

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

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

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

else if (t == '!') return 3;

else if (t == '\*' || t == '/' || t == '&') return 4;

else return 0;

}

void RPN(FILE\* f, char\* text)

{

int pR, pS, count = 0;

char buf;

char result[max], finalResult[max];

for (int i = 0; i <= strlen(text); i++)

{

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

while (tos > 0) {

result[count++] = ' ';

result[count++] = pop();

result[count++] = ' ';

}

result[count++] = '\n';

}

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

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

else if (tos == 0)

push(text[i]);

else {

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

pR = priorCheck(text[i]);

if (text[i] == '(')

push(text[i]);

else if (text[i] == ')')

while ((buf = pop()) != '(') {

result[count++] = ' ';

result[count++] = buf;

result[count++] = ' ';

}

else if (pS > pR)

push(text[i]);

else {

while (pR <= priorCheck(stack[tos - 1]) && tos != 0) {

result[count++] = ' ';

result[count++] = pop();

result[count++] = ' ';

}

push(text[i]);

}

}

}

result[count] = '\0';

char\* ptr = strtok(result, " ");

strcpy(finalResult, ptr);

while (ptr = strtok(NULL, " ")) {

strcat(finalResult, " ");

strcat(finalResult, ptr);

}

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

tos = 0;

}

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))

RPN(f2, text);

fclose(f1);

fclose(f2);

return 0;

}

**Результат работы программы.**

Входные данные:

Изображение выглядит как текст, клавиатура

Автоматически созданное описание

Выходные данные:

Изображение выглядит как стрела

Автоматически созданное описание

**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.