ГУО “БГУИР”

Факультет компьютерного проектирования

Кафедра Инженерной психологии и эргономики

Отчёт по

Лабораторной работе №4

СРЕДА ПРОГРАММИРОВАНИЯ VISUAL C++. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

Подготовил:

Студент гр.110101

Ладутько Я.Д.

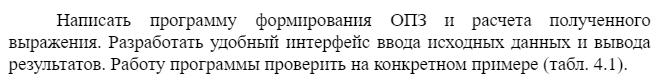
Проверила:

Семижон Е.А.

Минск 2022

***Цель работы:*** изучить правила формирования постфиксной записи арифметических выражений с использованием стека.

Вариант 16.





Код:

**///MySpace.h**

#pragma once

#include <iostream>

template <typename T>

struct stackData

{

T info = NULL;

stackData<T>\* next\_ptr = nullptr;

};

template <typename T>

class Stack

{

public:

stackData<T>\* \_top = nullptr;

~Stack()

{

while (\_top != nullptr)

pop();

}

T top()

{

T a = \_top->info;

return a;

}

void push(T x)

{

stackData<T>\* prew\_top = \_top;

if (\_top == nullptr)

{

\_top = new stackData<T>();

\_top->info = x;

}

else

{

\_top = new stackData<T>();

\_top->next\_ptr = prew\_top;

\_top->info = x;

}

}

void pop()

{

if (\_top != nullptr)

{

stackData<T>\* prew\_top = \_top;

\_top = \_top->next\_ptr;

delete prew\_top;

}

}

void outStack()

{

stackData<T>\* n\_top = \_top;

while (n\_top != nullptr)

{

std::cout << n\_top->info << std::endl;

n\_top = n\_top->next\_ptr;

}

}

};

template <typename T>

void input(T& a)

{

while (true)

{

std::cin >> a;

if (std::cin.fail() || (std::cin.peek() != '\n' && std::cin.peek() != ' ' && std::cin.peek() != '\t'))

{

std::cin.clear();

std::cin.ignore(10000, '\n');

std::cout << "Неверный ввод" << std::endl;

}

else

{

return;

}

}

}

void input(int& x, int a, int b);

void input(bool& b);

**///MySpace.cpp**

#include"MySpace.h"

using namespace std;

void input(int& x, int a, int b)

{

while (true)

{

input(x);

if (x < a || x > b)

{

cout <<

"Неверный ввод\n"

"Введите число на промежутке от " << a << " до " << b << endl;

}

else

return;

}

}

void input(bool& b)

{

char c;

while (true)

{

input(c);

if (c != 'y' && c != 'n')

{

cout <<

"Неверный ввод\n"

"Введите y или n\n";

}

else

{

b = (c == 'y' ? true : false);

return;

}

}

}

**///Lab4.cpp**

#include <iostream>

#include <random>

#include "MySpace.h"

#include <string>

using namespace std;

bool bracketsCheck(string s)

{

int x = 0;

for(char c : s)

{

if (c == '(')

x++;

else if (c == ')')

{

x--;

if (x < 0)

return false;

}

}

return x == 0;

}

bool isArifm(char c)

{

if (c == '+' || c == '-' || c == '/' || c == '\*')

return true;

return false;

}

bool stringCheck(string s)

{

for(char c : s)

{

if (!isalpha(c) && c != '(' && c != ')' && !isArifm(c))

return false;

}

char prew = '(';

for(int i = 0; i < s.length(); i++)

{

char c = s[i];

if(prew == '(')

{

if(isArifm(c))

return false;

if (c == ')')

return false;

}

else if(prew == ')')

{

if (isalpha(c))

return false;

if (c == '(')

return false;

}

else if(isalpha(prew))

{

if (c == '(')

return false;

if (isalpha(c))

return false;

}

else if(isArifm(prew))

{

if (c == ')')

return false;

if (isArifm(c))

return false;

}

if ((prew == '(' || prew == ')') && (s[i + 1] == '(' || s[i + 1] == ')'))

return false;

prew = c;

}

return true;

}

string removeSpaces(string s)

{

string new\_s;

for(char c : s)

{

if(c != ' ')

new\_s += c;

}

return new\_s;

}

bool compareSymbols(char a, char b)

{

if (a == '(')

{

return false;

}

else if (b == ')')

return true;

else if (a == '+' || a == '-')

{

if (b == '+' || b == '-')

return true;

return false;

}

else if (a == '\*' || a == '/')

{

if (b == '(' || b == ')')

return false;

return true;

}

}

int main()

{

setlocale(LC\_ALL, "RU");

string s, rez;

Stack<char> symbols;

while (true)

{

cout << "Введите выражение\n";

getline(cin, s);

s = removeSpaces(s);

if (!bracketsCheck(s) || !stringCheck(s))

{

cout << "Выражение записано неверно\n";

}

else

break;

}

s = '(' + s + ')';

symbols.push('(');

for(int i = 1; i < s.length(); i++)

{

char c = s[i];

if(isalpha(c))

{

rez += c;

}

else

{

if(compareSymbols(symbols.top(), c))

{

if (c == ')')

{

while (symbols.top() != '(')

{

rez += symbols.top();

symbols.pop();

}

symbols.pop();

}

else

{

while (compareSymbols(symbols.top(), c))

{

rez += symbols.top();

symbols.pop();

}

}

if (c != ')')

symbols.push(c);

}

else

{

symbols.push(c);

}

}

}

cout << "Обратная польская запись: " << rez << endl;

Stack <double> values;

double value;

for(char c : rez)

{

if(isalpha(c))

{

cout << "Введите " << c << endl;

double d;

input(d);

values.push(d);

}

else

{

switch (c)

{

case '+':

value = values.\_top->next\_ptr->info + values.top();

break;

case '-':

value = values.\_top->next\_ptr->info - values.top();

break;

case '\*':

value = values.\_top->next\_ptr->info \* values.top();

break;

case '/':

if(values.top() == 0)

{

cout << "Ошибка: деление на ноль\n";

return 0;

}

value = values.\_top->next\_ptr->info / values.top();

break;

}

values.pop();

values.pop();

values.push(value);

}

}

cout << "Ответ: " << values.top();

}

Вывод: изучил правила формирования постфиксной записи арифметических выражений с использованием стека.

