Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет инфомационных технологий и управления

Кафедра интеллектуальных информационных технологий

Основы алгоритмизации и программирования

Отчёт по лабораторной работе №5

**ОБРАТНАЯ ПОЛЬСКАЯ ЗАПИСЬ**

Студент гр. 321701 В. В. Перминова

Преподаватель С. И. Матюшкин

Минск 2024

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

Написать программу формирования ОПЗ и расчета полученного выражения. Разработать удобный интерфейс ввода исходных данных и вывода результатов. Работу программы проверить на конкретном примере.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер варианта | Выражение | a | b | c | d | e | Результат |
| 8 | a / (b ∙ (c + d)) – e | 8.5 | 0.3 | 2.4 | 7.9 | 1.6 | 1.151 |

Код программы:

#include <iostream>

#include <string>

#include <cctype>

#include <Windows.h>

using namespace std;

struct Stack {

string info;

Stack\* next;

} \*stack, \* solution;

void inStack(Stack\*\* p, string in) {//функция формирования элемента стека

Stack\* t = new Stack;

t->info = in;

t->next = \*p;

\*p = t;

}

string outStack(Stack\*\* p) {

string out;

Stack\* t = (\*p);

out = t->info;

(\*p) = (\*p)->next;

delete t;

return out;

}

string firstInStack(Stack\* p) {

if (p == NULL)

return "(";

else

return p->info;

}

void convert(Stack\*\* p, string\* in, string\* out) {

for (int i = 0; i <= (\*in).length(); i++) {

if (isdigit((\*in)[i])) {

string temp = "";

for (int j = i; isdigit((\*in)[j]) || (\*in)[j] == '.'; j++) {

temp.insert(temp.length(), 1, (\*in)[j]);

i = j;

}

(\*out) += temp;

}

else if ((\*in)[i] == '(')

inStack(&stack, "(");

else if ((\*in)[i] == ')') {

while (firstInStack(stack) != "(") {

\*out += outStack(&stack);

}

string temp;

outStack(&stack);

}

else if ((\*in)[i] == '+' || (\*in)[i] == '-' || (\*in)[i] == '\*' || (\*in)[i] == '/') {

string element = "";

if ((\*in)[i] == '+' || (\*in)[i] == '-') {

while (firstInStack(stack) != "(") {

string element = outStack(&stack);

(\*out).insert((\*out).length(), element);

}

string temp(1, (\*in)[i]);

inStack(&stack, temp);

}

else {

while (firstInStack(stack) != "(" && firstInStack(stack) != "+" && firstInStack(stack) != "-") {

string element = outStack(&stack);

(\*out).insert((\*out).length(), element);

}

string temp(1, (\*in)[i]);

inStack(&stack, temp);

}

}

}

while (stack != NULL) {

string element = outStack(&stack);

(\*out).insert((\*out).length(), element);

}

}

double Result(string str) {

double res;

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

if (isdigit(str[i])) {

string temp;

temp.push\_back(str[i]);

temp.push\_back(str[i + 1]);

temp.push\_back(str[i + 2]);

inStack(&solution, temp);

i = i + 2;

}

if (!isdigit(str[i])) {

double op1, op2;

op2 = stod(outStack(&solution));

op1 = stod(outStack(&solution));

if (str[i] == '+')

inStack(&solution, to\_string(op1 + op2));

if (str[i] == '-')

inStack(&solution, to\_string(op1 - op2));

if (str[i] == '\*')

inStack(&solution, to\_string(op1 \* op2));

if (str[i] == '/')

inStack(&solution, to\_string(op1 / op2));

}

}

res = stod(outStack(&solution));

return res;

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

string in = "";

string out = "";

cout << "Введите выражение: ";

getline(cin, in);

convert(&stack, &in, &out);

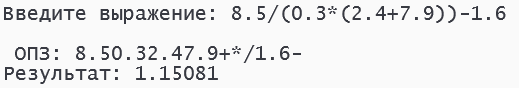
cout << "\n ОПЗ: " << out << endl;

cout << "Результат: " << Result(out) << endl;

return 0;

}

Результат:



Вывод: в ходе данной лабораторной работы мы изучили правила формирования постфиксной записи арифметических выражений с использованием стека.