**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: **Динамические структуры данных. Тестирование.**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6304 |  | Блинникова Ю.И. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Оглавление

[**Оглавление** 2](#_Toc483382275)

[Цель: 3](#_Toc483382276)

[Задание: 3](#_Toc483382277)

[Содержание: 3](#_Toc483382278)

[**Вывод** 5](#_Toc483382279)

Цель:

Написание программы с использованием динамических структур данных.

Задание:

**Стековая машина.**  
На вход программе подается последовательность (не более 100 элементов) из чисел и арифметических операций (+, -, \*, / (деление нацело)) разделенных пробелом, которые программа должна интерпретировать и выполнить по следующим правилам:

Если очередной элемент входной последовательности - число, то положить его в стек

Если очередной элемент - знак операции, то применить эту операцию над двумя верхними элементами стека, а результат положить обратно в стек (следует считать, что левый операнд выражения лежит в стеке глубже)

Если входная последовательность закончилась, то вывести результат (число в стеке)

Если в процессе вычисления возникает ошибка (для операции в стеке не хватает аргументов или по завершении работы программы в стеке более одного элемента), то вместо результата следует вывести "**error**"

Стек требуется реализовать самостоятельно на базе массива.

Содержание:

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

void push(int\* stack, int symbol, int\* i)//функция добавления элемента в стек

{

stack[\*i + 1] = symbol;

(\*i)++;

}

int pop(int\* i, int\* stack)//удаление элемента из стека

{

int cur = stack[\*i];

(\*i)--;

return cur;

}

int main()

{

char\* str = (char\*)malloc(sizeof(char) \* 100);

fgets(str, 100, stdin);

int count = -1;//последний элемент в стеке

int stack[100];//инициализация стека

char\*ptr = strtok(str, " ");//разбиение на отдельные лексемы

while (ptr)

{

int symbol = atoi(ptr);//конвертирует значение в величину типа int

if (symbol != 0)//если значение типа int

push(stack, symbol, &count);

else

{

//работа стековой машины

if (count < 1)//если в стеке есть 1 число или его вовсе нет

{

printf("error");

return 0;

}

int y = pop(&count, stack);//снимает элемент с вершины стека, перемещая вершину к следующему элементу

int x = pop(&count, stack);

switch (ptr[0])

{

case '+':

push(stack, x + y, &count);

break;

case'-':

push(stack, x - y, &count);

break;

case '\*':

push(stack, x \* y, &count);

break;

case '/':

push(stack, x / y, &count);

break;

default:

{

printf("error");

return 0;

}

}

}

ptr = strtok(NULL, " ");

}

int result = pop(&count, stack);

if (count > -1)//проверка на наличие элементов в стеке

{

printf("error");

return 0;

}

else

printf("%d", result);

return 0;

}

# Вывод

Выполнив лабораторную работу, использовав стандартные средства языка С, была самостоятельно смоделирована динамическая структура данных как стек на базе массива, которая позволила удобно обрабатывать значения в определенном порядке.