**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Программирование»**

Тема: **Стековая машина.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Иванов В. С. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

Содержание

[**Цель работы** 3](#_Toc480663232)

[**Задание** 3](#_Toc480663233)

[**Ход работы** 3](#_Toc480663234)

[**Вывод** 4](#_Toc480663235)

[**Приложение** 4](#_Toc480663236)

# **Цель работы**

Написание программы, выполняющей определённые действия, с помощью стека, реализованного на базе списка.

# **Задание**

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: 

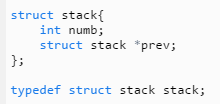
* Если очередной элемент входной последовательности - число, то положить его в стек;
* Если очередной элемент - знак операции, то применить эту операцию над двумя верхними элементами стека, а результат положить обратно в стек (следует считать, что левый операнд выражения лежит в стеке глубже);
* Если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

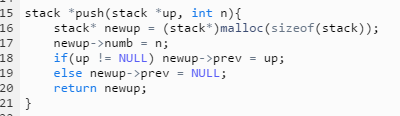
Стек требуется реализовать самостоятельно на базе списка.

# **Ход работы**

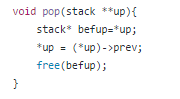
1. Создание структуры, описывающей стек.



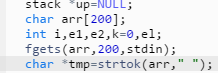
1. Добавление элемента в стек.



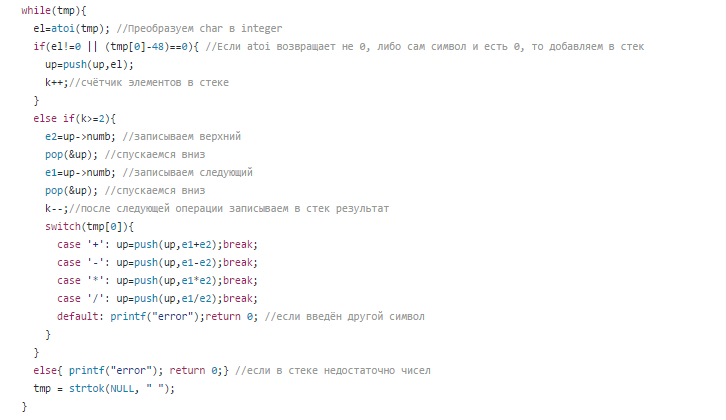
1. Удаление элемента из стека.



1. Объявление переменных.



1. Работа со стеком и выполнение операций.



1. Последняя проверка и вывод результата.



# **Вывод**

Выполнение данной лабораторной работы помогло разобраться с использованием стека и освоить на практике действия с ним, а также укрепить уже изученный материал.

# **Приложение**

|  |
| --- |
|  |
|  | //Выполнил Иванов Владимир 6304  //Задание представлено в отчёте |
|  |  |
|  | #include <stdio.h> |
|  | #include <stdlib.h> |
|  | #include <string.h> |
|  |  |
|  | struct stack{ |
|  | int numb; |
|  | struct stack \*prev; |
|  | }; |
|  |  |
|  | typedef struct stack stack; |
|  |  |
|  | stack \*push(stack \*up, int n){ |
|  | stack\* newup = (stack\*)malloc(sizeof(stack)); |
|  | newup->numb = n; |
|  | if(up != NULL) newup->prev = up; |
|  | else newup->prev = NULL; |
|  | return newup; |
|  | } |
|  |  |
|  | void pop(stack \*\*up){ |
|  | stack\* befup = \*up; |
|  | \*up=(\*up)->prev;  free(befup); |
|  |  |
|  | } |
|  |  |
|  | int main(){ |
|  | stack \*up=NULL; |
|  | char arr[200]; |
|  | int i,e1,e2,k=0,el; |
|  | fgets(arr,200,stdin); |
|  | char \*tmp=strtok(arr," "); //Берём символ(число) до пробела |
|  | while(tmp){ |
|  | el=atoi(tmp); //Преобразуем char в integer |
|  | if(el!=0 || (tmp[0]-48)==0){ //Если atoi возвращает не 0, либо сам символ и есть 0, то добавляем в стек |
|  | up=push(up,el); |
|  | k++;//счётчик элементов в стеке |
|  | } |
|  | else if(k>=2){ |
|  | e2=up->numb; //записываем верхний |
|  | pop(&up); //спускаемся вниз |
|  | e1=up->numb; //записываем следующий |
|  | pop(&up); //спускаемся вниз |
|  | k--;//после следующей операции записываем в стек результат |
|  | switch(tmp[0]){ |
|  | case '+': up=push(up,e1+e2);break; |
|  | case '-': up=push(up,e1-e2);break; |
|  | case '\*': up=push(up,e1\*e2);break; |
|  | case '/': up=push(up,e1/e2);break; |
|  | default: printf("error");return 0; //если введён другой символ |
|  | } |
|  | } |
|  | else{ printf("error"); return 0;} //если в стеке недостаточно чисел |
|  | tmp = strtok(NULL, " "); |
|  | } |
|  | if(k==1) printf("%d",up->numb); //проверка на наличие в стеке лишних элементов |
|  | else printf("error"); |
|  | return 0; |
|  | } |