**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №6**

**по дисциплине «Программирование»**

Тема: Моделирование стека

|  |  |  |
| --- | --- | --- |
| Студент гр. 6304 |  | Запевалов А.И. |
| Преподаватель |  | Берленко Т.А. |

Санкт-Петербург

2017

**Цель работы**

Необходимо реализовать программу, которая моделирует работу стека на базе массива.

**Задание**

Моделирование стека.

Требуется написать программу, моделирующую работу стека, реализовав перечисленные ниже методы. Программе на вход подается последовательность команд с новой строки (не более 100 команд), в зависимости от которых программа выполняет ту или иную операцию и выводит результат ее выполнения с новой строки.

Перечень команд:

push n - добавляет целое число n в стек. Программа должна вывести "ok"

pop - удаляет из стека последний элемент и выводит его значение на экран

top - программа должна вывести верхний элемент стека на экран не удаляя его из стека

size - программа должна вывести количество элементов в стеке

exit - программа должна вывести "bye" и завершить работу

Если в процессе вычисления возникает ошибка (например вызов метода pop при пустом стеке), программа должна вывести "error" и завершиться.

Стек требуется реализовать самостоятельно на базе массива.

Исходный код программы:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int push(int\* array, int j, char\* a) //adds new element

{

array[j]=atoi(a);

printf("ok\n");

return j+1;

}

void size(int j) //shows current index

{

printf("%d\n", j);

}

int printexit()//ends program

{

printf("bye\n");

return 1;

}

int pop(int\* array, int j)//prints and deletes top element

{

printf("%d\n", array[j-1]);

return j-1;

}

void top(int\* array, int j)//shows top element

{

printf("%d\n", array[j]);

}

int main()

{

int i=0, c=1, n, j=0, flag=0; // c - input strings counter, n - result strings amount counter

char del[]="\n "; // i - strtok counter, j - result counter

char vvod[200][20]; // del - dividing symbols array, vvod - input array

char mas[200][20]; // mas - in-between result counter

int array[200]; // array - result counter

vvod[0][0]='0'; //checking previous result needs array to begin with 1

while(strcmp(vvod[c-1], "exit\n")!=0) //reading strings

{

fgets(vvod[c], 20, stdin);

//printf("%s\n", vvod[c]);

c++;

}

for (n=1; n<c; n++)

{

char \*p=strtok(vvod[n], del); // Divides words from input strings

while (p != NULL)

{

strcpy(mas[i], p);

p = strtok (NULL, del);

i++;

}

}

for (n=0; n<i; n++)

{

if (flag==0)

{

if (strcmp(mas[n], "push")==0)

{

j=push(array, j, mas[n+1]);

n++; //jumps over the number

}

else if (strcmp(mas[n], "size")==0)

{

size(j);

}

else if (strcmp(mas[n], "exit")==0)

{

flag=printexit();

}

else if (j!=0)

{

if (strcmp(mas[n], "pop")==0)

{

j=pop(array, j);

}

else if (strcmp(mas[n], "top")==0)

{

top(array, j-1);

}

}

else

{

printf("error\n");

flag=1;

}

}

}

return 0;

}

Введенные команды разделяются на «слова» пробелами и сохраняются в массиве строк.

char \*p=strtok(vvod[n], del); // Divides words from text into strings

while (p != NULL)

{

strcpy(mas[i], p);

p = strtok (NULL, del);

i++;

}

Выполняются действия в зависимости от введенных команд: размер численно равен индексу текущего элемента, добавление элемента увеличивает индекс текущего элемента на 1, удаление уменьшает на 1.

**Вывод.**

Язык С позволяет создавать и обрабатывать массивы подобно стеку – вносить и удалять элементы, считывать их.