

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

Курков Д.В.

Преподаватель

Берленко Т.А.

Санкт-Петербург

2017

Цель.

На базе массива написать программу, моделирующую работу стека.

Задание.

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.

Ход работы.

1. Создание стека

```
int *stack;  
int t = 0;
```

2. Реализация функций для работы со стеком

- Функция для добавления элемента в стек

```
void push (int n)  
{  
    stack = (int*)realloc(stack, sizeof(int)*(t+1));  
    *(stack+(t++)) = n;  
    printf ("ok\n");  
}
```

- Функция для считывания элемента из стека

```
void pop ()  
{  
    if (!t)  
        error();  
    printf ("%d\n", *(stack+(--t)));  
}
```

- Функция для получения верхнего элемента

```
void top () //получить верхний элемент  
{  
    if (!t)  
        error();  
    printf ("%d\n", *(stack+t-1));  
}
```

- Функция для получения размера

```
void size ()
{
    printf ("%d\n", t);
}
```

- Ошибка!

```
void error ()
{
    printf ("error\n");
    exit (0);
}
```

3. Создание main для работы с вышеперечисленными функциями.

```
int main ()
{
    stack = (int*)malloc(sizeof(int));
    char *cmds = (char*)malloc(sizeof(char));
    char *cmd = (char*)malloc(sizeof(char));
    do
    {
        int i = 0;
        while ((*cmd+(i++)) = getchar()) != '\n')
            cmd = (char*)realloc(cmd, sizeof(char)*(i+1));
        *(cmd+i) = '\0';
        cmds = (char*)realloc(cmds, (strlen(cmds)+strlen(cmd))*sizeof(char)+1);
        strcat(cmds, cmd);
    }while (strcmp(cmd, "exit\n"));
    cmd = strtok (cmds, " \n");
    do
    {
        if (!(strcmp(cmd, "push")))
        {
            cmd = strtok (NULL, " \n");
            for (int i = (*cmd == '-') ? 1 : 0; *(cmd+i) != '\0'; i++)
                if (!isdigit(*(cmd+i)))
                    error();
            push(atoi(cmd));
        }
        else if (!(strcmp(cmd, "pop")))
            pop();
        else if (!(strcmp(cmd, "top")))
            top();
        else if (!(strcmp(cmd, "size")))
            size();
        else if (strcmp(cmd, "exit"))
            error();
    }while (cmd = strtok(NULL, " \n"));
    printf ("bye");
    return 0;
}
```

Вывод: В ходе работы был реализован стек на базе массива. Были получены навыки взаимодействия со стеком.