

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

```
int pop ()  
{  
    if (!t)  
        return 1;  
    printf ("%d\n", *(stack+(--t)));  
    return 0;  
}
```

- Функция для получения верхнего элемента

```
int top ()  
{  
    if (!t)  
        return 1;  
    printf ("%d\n", *(stack+t-1));  
    return 0;  
}
```

- Функция для получения размера

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

3. Создание main для работы с вышеперечисленными функциями.

```
int main ()
{
    stack = (int*)malloc(sizeof(int));
    char cmds[5000];
    char *cmd = (char*)malloc(sizeof(char)*30);
    do
    {
        fgets(cmd, 30, stdin);
        strcat(cmds, cmd);
    } while (strcmp(cmd, "exit\n"));
    int err_f = 0;
    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)))
                    err_f = 1;
            if (!err_f)
                push(atoi(cmd));
        }
        else if (!(strcmp(cmd, "pop")))
            err_f = pop();
        else if (!(strcmp(cmd, "top")))
            err_f = top();
        else if (!(strcmp(cmd, "size")))
            size();
        else if (strcmp(cmd, "exit"))
            err_f = 1;
    } while ((cmd = strtok(NULL, " \n")) && !err_f);
    (err_f) ? printf("error") : printf("bye");
    free(stack);
    return 0;
}
```

4. Программа протестирована, ее вывод проверен на корректность.

Вывод: В ходе работы было проведено знакомство с таким типом хранения данных, как стек, получены навыки взаимодействия с ним, реализован стек на базе массива.