

	<p align="center"> <b>Министерство образования и науки Российской Федерации</b>  <b>Федеральное государственное бюджетное образовательное</b>  <b>учреждение</b>  <b>высшего образования</b>  <b>«Московский государственный технический университет</b>  <b>имени Н.Э. Баумана</b>  <b>(национальный исследовательский университет)»</b>  <b>(МГТУ им. Н.Э. Баумана)</b> </p>
---	--

---

ФАКУЛЬТЕТ \_\_\_\_\_ Информатика и системы управления  
 КАФЕДРА \_\_\_\_\_ Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 4**  
**По дисциплине «Типы и структуре данных»**

Название **«Работа со стеком»**

Студент Дубов Андрей Игоревич  
 фамилия, имя, отчество

Группа ИУ7-33Б

Вариант 5

Тип лабораторной работы Учебная

Студент	_____	<u>Дубов А. И.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Преподаватель	_____	<u>Рыбкин Ю. А.</u>
	<i>подпись, дата</i>	<u>Силантьева А. В.</u>
		<i>фамилия, и.о.</i>

2022 г.

# Оглавление

- Условие задачи.....3
- Описание технического задания.....3
- Входные данные:.....3
- Выходные данные:.....3
- Аварийные ситуации:.....3
- Описание структуры данных.....3
- Описание алгоритма.....4
- Набор тестов.....4
- Оценка эффективности.....4
- Вывод.....5
- Ответы на контрольные вопросы.....5

## Условие задачи

Разработать программу работы со стеком, реализующую операции добавления и удаления элементов из стека и отображения текущего состояния стека. Реализовать стек: а) массивом; б) списком.

Все стандартные операции со стеком должны быть оформлены отдельными подпрограммами. В случае реализации стека в виде списка при отображении текущего состояния стека предусмотреть возможность просмотра адресов элементов стека и создания дополнительного собственного списка свободных областей (адресов освобождаемой памяти при удалении элемента, который можно реализовать как списком, так и массивом) с выводом его на экран.

## Описание технического задания

Проверить правильность расстановки скобок трех типов (круглых, квадратных и фигурных) в выражении.

## Входные данные:

1. Команда для вызова функции
2. Данные запрашиваемые программой

## Выходные данные:

1. Уведомление о выполнении команды
2. Результаты проверки строки

## Аварийные ситуации:

1. Некорректный ввод номера команды.

## Описание структуры данных

### Структура стека

```
typedef struct
{
    int *data;
    size_t size;
    size_t top;
} stack_t;
```

data – указатель на данные

size – размер выделенной памяти

top – указатель на последний элемент

### Структура списка

```
typedef struct node_t
{
    int value;
    struct node_t *next;
} node_t;
```

value – элемент

next – указатель на следующий элемент

## Описание алгоритма

1. Пользователю предлагается режим
2. Пользователь управляет выделенной памятью
3. Пока пользователь не введет 0 (выход из программы), ему будет предложено вводить номера команд и выполнять действия по выбору.

## Набор тестов

	Название теста	Пользователь вводит	Вывод
1	Некорректный ввод команды	45	No such option or wrong input
2	Пустой ввод	Пустой ввод.	No such option or wrong input
3	Команда 1\2 1	000	OK
4	Команда 1\2 1	0)))	NOT OK
5	Команда 1\2 3 (стек пуст)		Stack empty
6	Команда 1\2 2	Число	Добавлен элемент
7	Команда 1\2 3		Удалён элемент
8	Команда 1\2 5		Очищает стек
9	Команда 0	0	Выход из программы

## Оценка эффективности

	Добавление (тактов)	Удаление (тактов)
Список	1467	695
Массив	5308	3255

	Память (байт)
Список	1600 (на 100 элементов)

Массив	664 (на 160 выделенных элементов)
--------	-----------------------------------

## Вывод

На добавление элементов в стек и удаление элементов из стека затрачивается одно и то же время. Добавление/удаление элементов из стека, организованного как статический массив, происходит на 70 процентов быстрее, чем добавление/удаление элементов из стека, организованного как список. При известном кол-ве элементов стек, организованный как статический массив, занимает на 70 процентов меньше памяти, чем стек-список, так как в списке кроме данных приходится хранить указатель на следующий элемент. Однако при работе программы кол-во элементов в стеке меняется и при заполненности стека меньше половины от максимального кол-ва элементов по памяти выгоднее использовать список. В ходе тестирования программы, было выявлено, что фрагментация данных стека происходит не всегда.

## Ответы на контрольные вопросы

### 1. Что такое стек?

Стек - это структура данных, в которой можно обрабатывать только последний (верхний) элемент. Стек подчинён правилу LIFO (последний пришёл - первый вышел).

### 2. Каким образом и сколько памяти выделяется под хранение стека при различной его реализации?

При хранении стека как списка, память выделяется в куче при добавлении элемента (выделяется столько памяти, сколько необходимо для хранения одного элемента). При хранении стека как массива, память выделяется сразу вся, либо на стеке (в случае статического массива), либо в куче (в случае динамического массива). Для каждого элемента списка как стека выделяется дополнительная память на хранение указателя на следующий элемент. (либо 4, либо 8 байт, в зависимости от разрядности процессора).

### 3. Каким образом освобождается память при удалении элемента стека при различной реализации стека?

При хранении стека как списка, при удалении верхнего элемента смещается указатель, который указывает на начало стека и освобождается память, выделенная под удалённый элемент. При хранении стека как массив, вся память освобождается при завершении программы.

### 4. Что происходит с элементами стека при его просмотре?

Элементы стека удаляются.

##### 5. Каким образом эффективнее реализовывать стек? От чего это зависит?

Если необходимо существенное преимущество во времени, а объём памяти, выделяемый под стек, не интересует, то намного эффективнее обрабатывать стек как массив. Если требуется "универсальность" (ведь в случае статического массива наблюдается ограниченность в ресурсах, а в случае динамического массива наблюдаются существенные затраты на перевыделение памяти) памяти, выделяемой под стек, то удобнее обрабатывать стек как список.