|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство образования и науки Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3  
«Работа со стеком»**

Студент, группа **Пронин А.С. ИУ7-32Б**

*2020 г.*

# Условия задачи

Создать программу работы со стеком, выполняющую операции добавления, удаления элементов и вывод текущего состояния стека.

Реализовать стек:

* массивом;
* списком.

Все стандартные операции со стеком должны быть оформлены подпрограммами. При реализации стека списком в вывод текущего состояния стека добавить просмотр адресов элементов стека и создать свой список или массив свободных областей (адресов освобождаемых элементов) с выводом его на экран.

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

# Техническое задание

## Входные данные

1) Целое число от 0 до 14 (для ввода номера команды из меню)

2) Строка (выражение для ввода стека)

3) Целое число до 10000 (для ввода максимального кол-ва элементов)

4) Символ (для добавления элемента в стек)

5) Целое число от 1 до 50000 (для ввода количества итераций для тестов на время выполнения программы)

6) Целое число (для ввода кол-ва элементов в стеке для оценки объёма занимаемой памяти разными структурами данных)

## Выходные данные

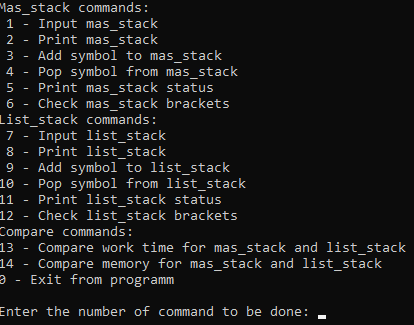
Вывод стека (для masstack и liststack)

Время информации о стеке (для masstack)

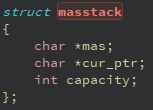
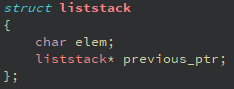
Вывод адресов элементов в стеке (для liststack)

## Функции программы

Программа представляет из себя консольное приложение исполняющее следующие команды:



## Структуры данных

# Описание алгоритма

Команды:

1 – Вводит стек в виде массива

2 – Выводит стек в виде массива

3 – Добавляет элемент в стек в виде массива

4 – Удаляет элемент из стека в виде массива, а также печатает его.

5 – Выводит информацию о стеке в виде массива

6 – Проверяет правильность расставленных скобок в стеке в виде массива

7 – Вводит стек в виде списка

8 – Выводит стек в виде списка

9 – Добавляет элемент в стек в виде списка

10 – Удаляет элемент из стека в виде списка, а также печатает его адрес и сам элемент.

11 – Выводит адреса элементов в стеке в виде списка

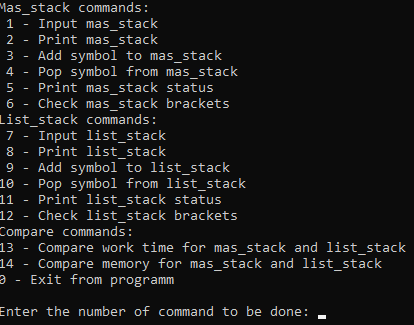
12 – Проверяет правильность расставленных скобок в стеке в виде списка

13 – Выводит время выполнения программы для стеков разных видов

14 – Выводит объёмы занимаемой памяти для стеков разных видов

# Тесты

### Вот так выглядит меню:



### Некорректный ввод:

1. Некорректная команда





1. Команда 1/7 (Пустой ввод)



1. Команда 3/9 (Переполнение)



1. Команда 4/6/10/12 (Пустой стек)



1. Команда 13



1. Команда 14



# Оценка эффективности

## Память

## Эффективность

# Контрольные вопросы

**1.** **Что такое стек?**

Стек – структура данных, в которой можно обрабатывать только последний добавленный элемент (верхний элемент). На стек действует правило LIFO — последним пришел, первым вышел.

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

При хранении стека с помощью списка, то память всегда выделяется в куче. При хранении с помощью массива, память выделяется либо в куче, либо на стеке (в зависимости от того, динамический массив или статический). Для каждого элемента стека, реализованного списком, выделяется на 4 или 8 байт (на большинстве современных ПК) больше, чем для элемента массива. Эти дополнительные байты занимает указатель на следующий элемент списка. Размер указателя (4 или 8 байт) зависит от архитектуры.

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

При хранении стека связанным списком, верхний элемент удаляется освобождением памяти для него и смещением указателя, указывающего на начало стека. При удалении из стека, реализованного массивом, смещается лишь указатель на вершину стека.

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

Элементы стека уничтожаются, так как каждый раз достается верхний элемент стека.

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

Реализовывать стек эффективнее с помощью массива. Он выигрывает как во времени обработки, так и в количестве занимаемой памяти (в классическом случае). Вариант хранения списка может выигрывать только в том случае, если стек реализован статическим массивом. В этом случае, память для списка ограничена размером оперативной памяти (так как память выделяется в куче), а память для статического массива ограничена размером стека.

# Вывод