|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования 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)** |

Отчет по лабораторной работе № 4

«Работа со стеком»

Работу выполнил:

студент группы ИУ7-32Б

Лемешев А.П.  
Работу проверил:  
Силантьева А.В.

Москва, 2021 г.

**Условие задачи**

Разработать программу работы со стеком, реализующую операции добавления и удаления элементов из стека и отображения текущегосостояния 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. Список свободных областей необходим для того, чтобы проследить, каким образом происходит выделение памяти менеджером памяти при запросах на нее и убедиться в возникновении или отсутствии фрагментации памяти.

**Техническое задание**

Ввести арифметическое выражение типа:

число|знак| ... число|знак|число| (где знак — это знак «+» или «-» или «\*» или «/»)

Числа целые. Вычислить значение выражения

**Входные данные**

Номер команды, размер стека, количество вводимых значений, значения типа int или выражение из условия задания.

**Выходные данные**

Элементы стека, список адресов, либо время выполнения, затрачиваемое на разное количество операций, также области освобожденной памяти в виде односвязного списка, и занимаемая память.

**Функции программы**

* Выйти из программы
* Заполнить стек (список)
* Добавить элемент в стек (список)
* Удалить элемент из стека (список)
* Текущее состояние стека (список)
* Вычислить с помощью стека (список)
* Свободные адреса стека (список)
* Заполнить стек (массив)
* Добавить элемент в стек (массив)
* Удалить элемент из стека (массив)
* Текущее состояние стека (массив)
* Вычислить с помощью стека (массив)
* Статистика по времени и памяти

**Возможные аварийные ситуации**

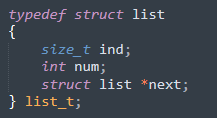
* Неверный пункт меню
* Неверный символ в выражении
* Первый или последний символ в выражении не число
* Размер стека указан неверно
* Количество вводимых элементов указано неверно
* Переполнение стека
* Ошибка выделения памяти
* Удаление элемента при пустом стеке

**Способ обращения к программе**

Запуск через командную оболочку MSYS2 (Windows), терминал sh/bash (Linux).

**Структуры данных**

**Стек, реализованный в виде списка**



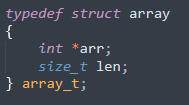
Поля структуры:

1. ind — номер элемента в стеке

2. num — значение текущего элемента в стеке

3. next — указатель на следующий элемент стека

**Стек, реализованный в виде массива**



Поля структуры:

1. arr — указатель на текущий элемент стека

2. len — количество элементов в стеке

# **Описание алгоритма**

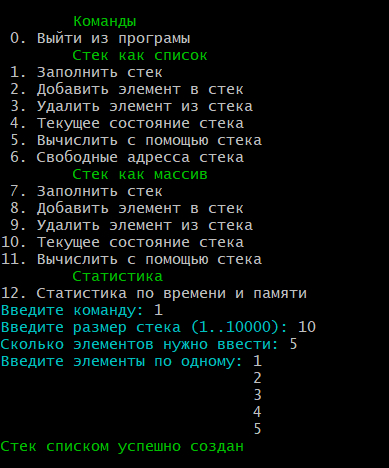
1. Выводится меню программы

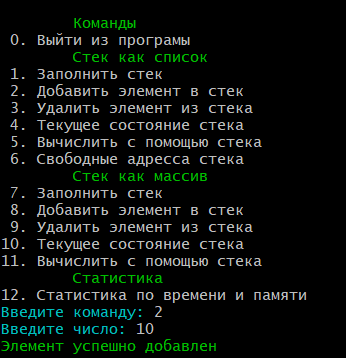
2. Пользователь вводит номер любой команды, которой соответствует свое назначение

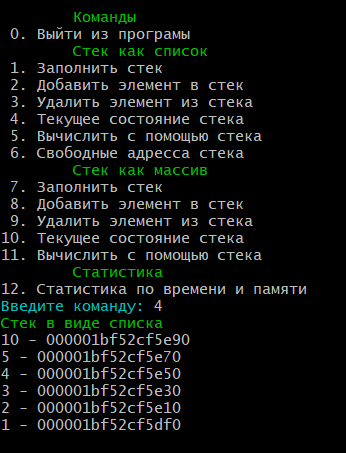
3. Ввод осуществляется, пока не будет совершена ошибка при вводе (аварийная ситуация) или пока не будет введен 0 (означает выход из программы)

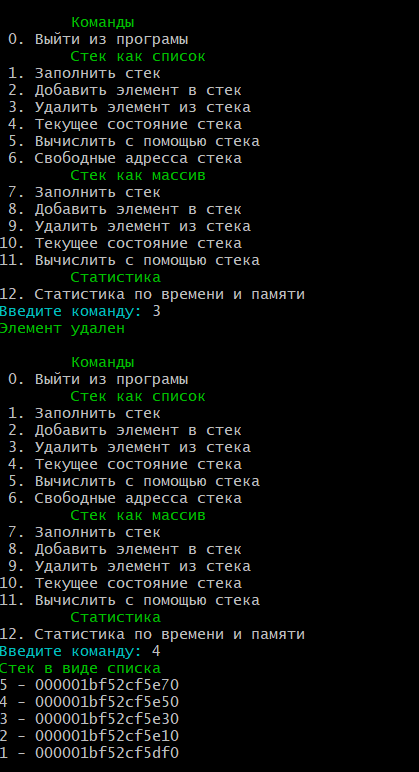
**Тесты**

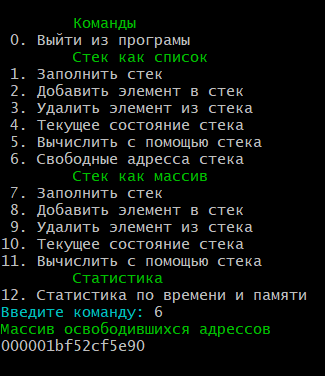
**Позитивные тесты**

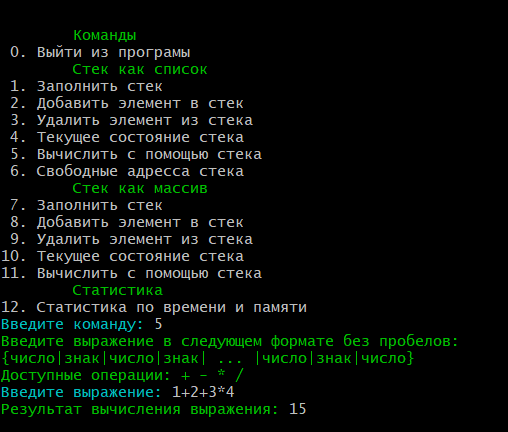
****

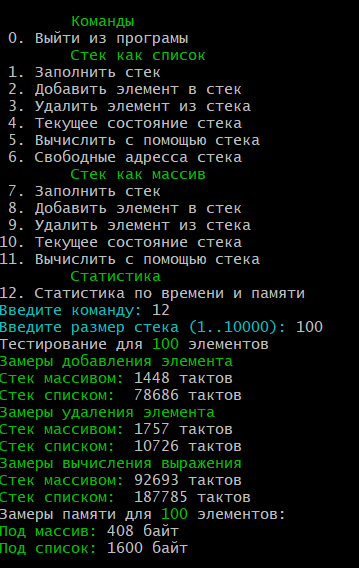
****

****

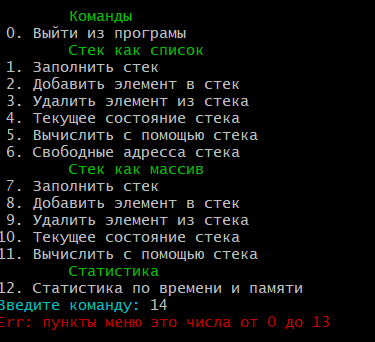
****

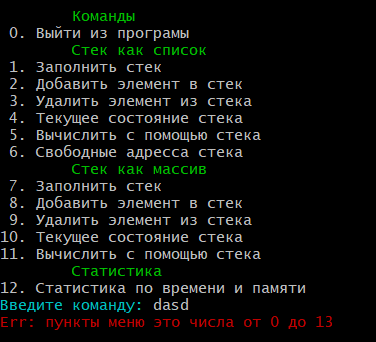
****

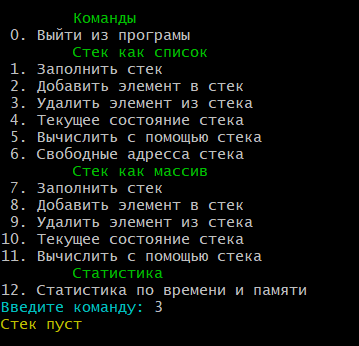
****

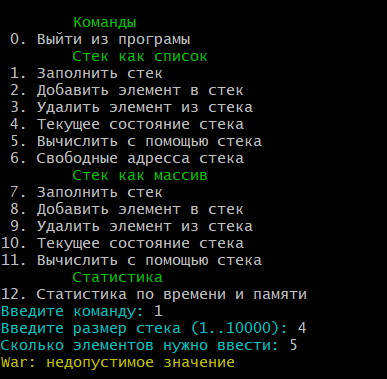
****

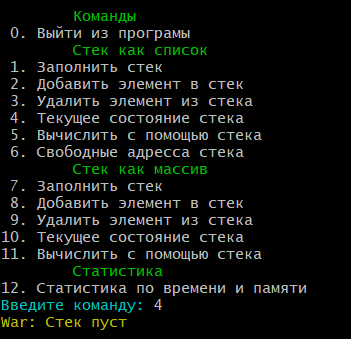
**Негативные тесты**

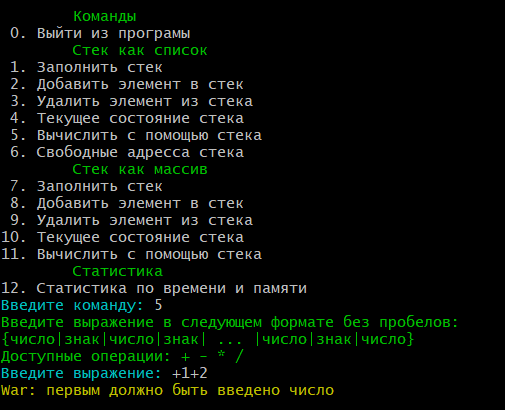
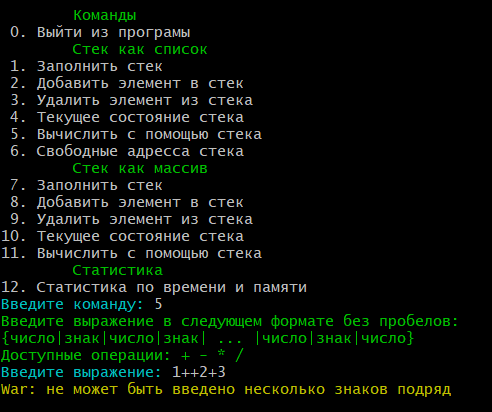
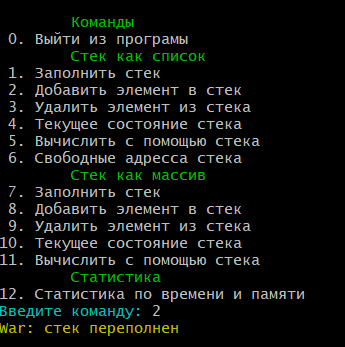












**Оценка эффективности**

Добавление элементов (в тактах)

|  |  |  |
| --- | --- | --- |
| Размер | Массив | Список |
| 10 | 121 | 2324 |
| 100 | 1097 | 18918 |
| 500 | 5306 | 88523 |
| 1000 | 9795 | 183393 |

Удаление элементов (в тактах)

|  |  |  |
| --- | --- | --- |
| Размер | Массив | Список |
| 10 | 158 | 1404 |
| 100 | 943 | 9559 |
| 500 | 3170 | 22612 |
| 1000 | 6331 | 44240 |

Вычисление выражения (в тактах)

|  |  |  |
| --- | --- | --- |
| Размер | Массив | Список |
| 10 | 9130 | 12185 |
| 100 | 63120 | 95178 |
| 500 | 231616 | 437806 |
| 1000 | 450726 | 858698 |

Замеры памяти (в байтах)

|  |  |  |
| --- | --- | --- |
| Размер | Массив | Список |
| 10 | 48 | 160 |
| 100 | 408 | 1600 |
| 500 | 2008 | 8000 |
| 1000 | 4008 | 16000 |

**Ответы на контрольные вопросы**

*1. Вопрос: что такое стек?*  
**Ответ:** последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны – с вершины стека. Стек функционирует по принципу: последним пришел – первым ушел, Last In – First Out (LIFO).

*2. Вопрос: каким образом и сколько памяти выделяется под хранение стека при различной его реализации?***Ответ:** в случае реализации стека в виде массива (вектора) выделяется непрерывная область памяти ограниченного размера, имеющая нижнюю и верхнюю границу, а также память под указатель на верхний элемент стека и текущий элемент. Размер статического массива известен на этапе трансляции, а динамического – на этапе выполнения программы. В случае реализации стека в виде односвязного линейного списка память выделяется последовательно под указатель на предыдущий узел списка и под значение текущего элемента стека.

*3. Вопрос: каким образом освобождается память при удалении элемента стека при различной реализации стека?*  
**Ответ:** при реализации стека в виде статического массива память не освобождается. При удалении элемента сначала считываются данные, а затем происходит перемещение указателя на верхний элемент стека к предыдущему элементу. Если указатель стека выходит за нижнюю границу массива, то стек пуст. То же самое происходит и в случае динамического массива, но перед завершением программы нужно освободить область памяти, на которую указывает указатель-начало массива. В случае списка сначала по указателю стека считывается информация об исключаемом элементе, а затем указатель смещается к предыдущему элементу. После чего освобождается память, выделенная под элемент. Если указатель имеет значение нулевого адреса, то стек пуст.

*4. Вопрос: что происходит с элементами стека при его просмотре?*  
**Ответ:** классическая реализация стека предполагает, что просмотреть содержимое стека без извлечения его элементов невозможно.

*5. Вопрос: каким образом эффективнее реализовывать стек? От чего это зависит?*  
**Ответ:** для ответа на этот вопрос я провел анализ эффективности работы со стеком двух различных реализаций по времени и памяти. Заметим, что вне зависимости от кол-ва элементов и операция добавления элемента в стек, и операция удаления выполняются быстрее в случае работы с массивом (примерно в 9-11 раз). Кроме того, список в худшем случае будет занимать в 16 раз больше памяти, чем массив. А в наилучшем – 16 байт. Таким образом, для работы со стеком небольшого объема эффективнее использовать односвязный линейный список. А при неопределенном (или большом) количестве элементов целесообразнее использовать массив.

# **Вывод**

В случае необходимости реализации стека на компьютере, его стоит реализовывать на динамическом массиве, так он имеет преимущества над статическим массивом и списком

Статический массив ограничен по памяти, так как располагается в стеке самого компьютера, в то время как динамический массив ограничен лишь объемом оперативной памяти компьютера

Список, в свою очередь, занимает больше количества памяти (примерно в 4 раза), так как ему, помимо самого элемента, нужно хранить указатель на следующий элемент списка, а также скорость работы с элементами списка занимает больше времени:

при удалении (примерно в 10 раза)

при добавлении (примерно в 20 раз)

при подсчете выражения (примерно в 1.5 раза)