|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_Программное обеспечение ЭВМ и информационные технологии\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

**“Работа со стеком”**

**Вариант 6**

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гурова Наталия Алексеевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_\_ИУ7-34Б\_\_\_\_\_\_\_\_\_

Выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_Гурова Н.А\_\_\_\_\_

*подпись, дата фамилия, и.о.*

Принял \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Барышникова М. Ю. \_\_\_

*подпись, дата фамилия, и.о.*

*2021 г.*

**Цель работы**

Цель работы - реализовать операции работы со стеком, который представлен в виде массива (статического или динамического) и в виде односвязного линейного списка; оценить преимущества и недостатки каждой реализации: получить представление о механизмах выделения и освобождения памяти при работе со стеком.

**Задание**

Создать программу работы со стеком, выполняющую операции добавление, удаления элементов и вывод текущего состояния стека. Реализовать стек:

а) массивом;

б) 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.

Используя стек, определить, является ли строка палиндромом.

**Входные данные**

Для создания стеков необходимо ввести максимальное количество элементов в стеках. Для добавления элемента (элементов) в стеки необходимо ввести соответствующий символ (строку). Для удаления элементов необходимо ввести количество этих элементов.

**Выходные данные**

При выборе пункта меню номер 4 будет выведено актуальное состояние стеков, пункт 5 печатает ответ на вопрос, являются ли соответствующие стеки палиндромами, пункт номер 6 выводит освобожденные адреса, пункт номер 7 позволяет провести исследование по использованию стеками памяти и времени при различных размерностях стеков.

**Способ обращения к программе**

Программа может быть вызвана через консоль с помощью команды app.exe

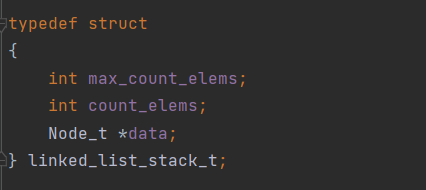
**Аварийные ситуации**

Могут быть выведены такие ошибки, как:

* Неверный введенный максимальный размер стеков
* Попытка вызвать пункт меню, использующийся для обработки массива, не создав массив
* Попытка добавить элемент в уже заполненный стек
* Попытка удалить элемент из пустого стека
* Неверно введен пункт из меню

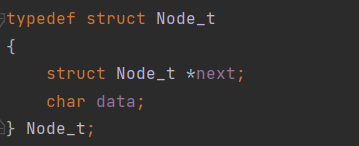
**Структуры данных**

Для хранения стека в виде односвязного списка использовалась следующая структура:

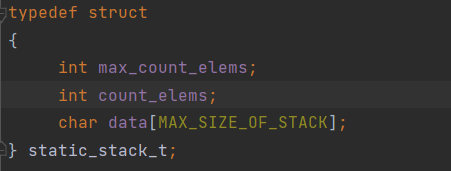
 Где:

* max\_count\_elems – максимальное количество элементов в стеке
* count\_elems – текущее количество элементов в стеке
* data – элементы стека

Для хранения информации об элементах в стеке в виде односвязного списка использовался односвязный список, состоящий из следующих узлов:

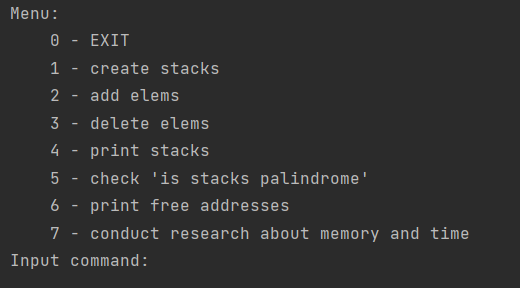
* next – указатель на следующий узел в списке
* data - значение соответствующего узла

Для хранения стека в виде статического массива использовалась следующая структура:

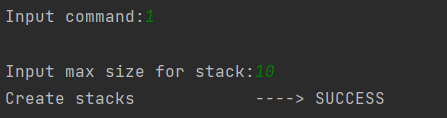
* max\_count\_elems – максимальное количество элементов в стеке
* count\_elems – текущее количество элементов в стеке
* data – элементы стека

**Описание алгоритма**

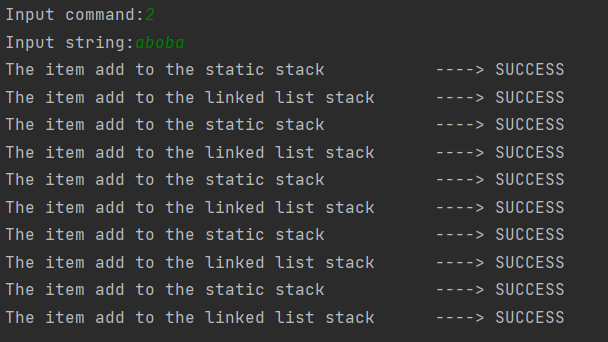
Данная программа предназначена консольное приложение со следующими возможными операциями, представленными в меню:



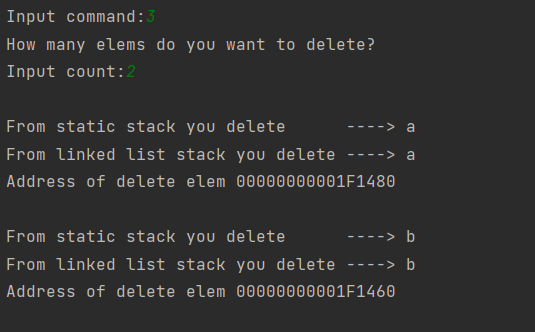
Пример создания стеков:



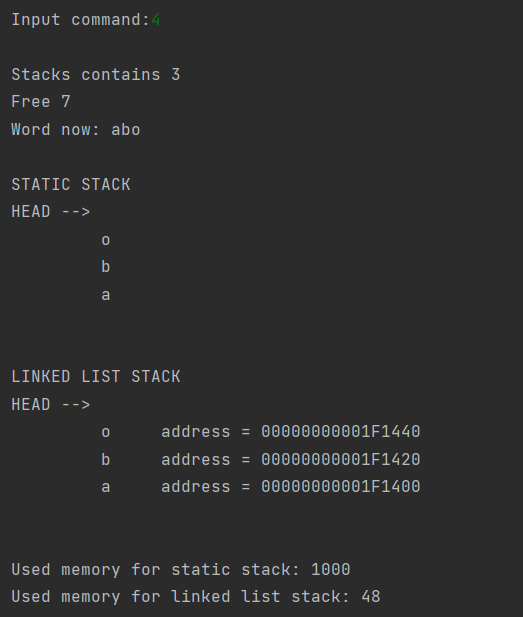
Пример добавления элементов:



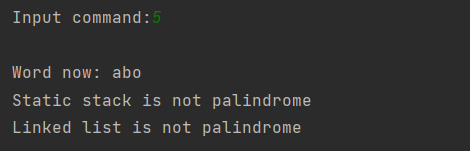
Пример удаления элементов:



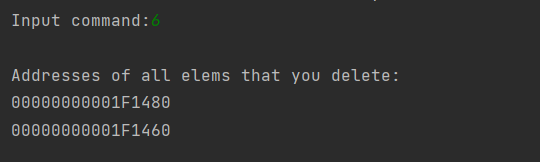
Пример вывода информации о стеках:



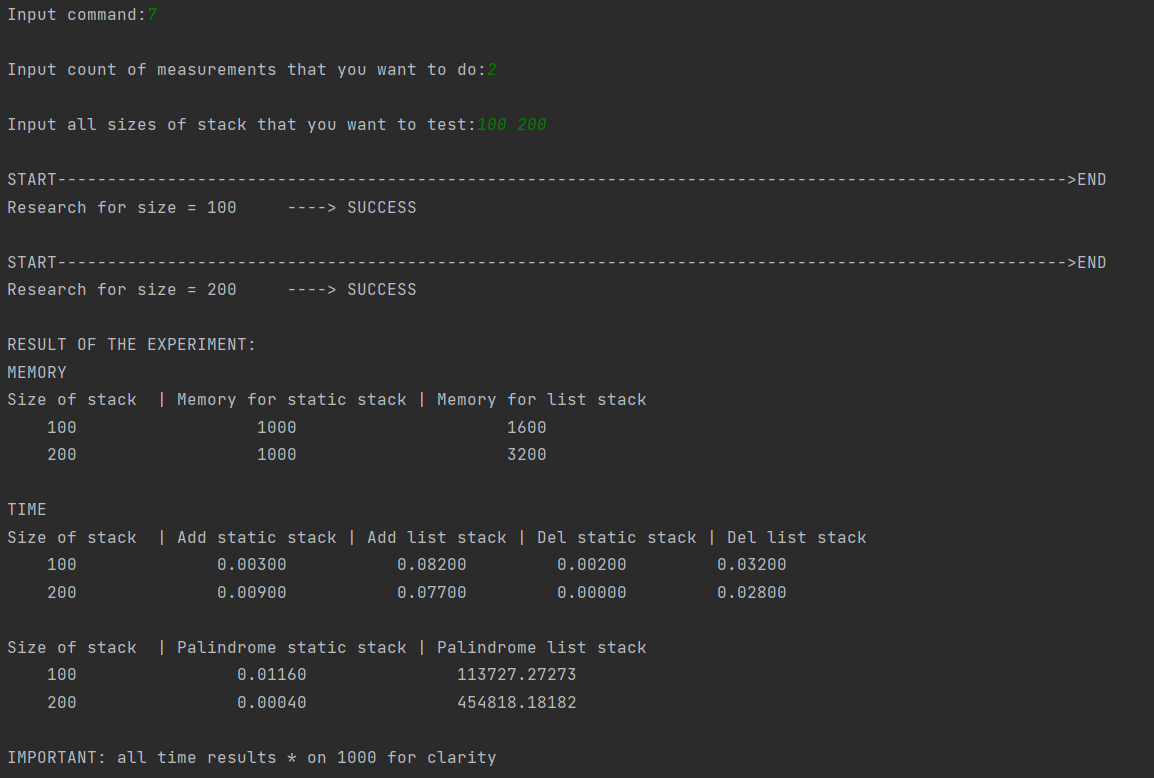
Пример проверки на палиндромность:



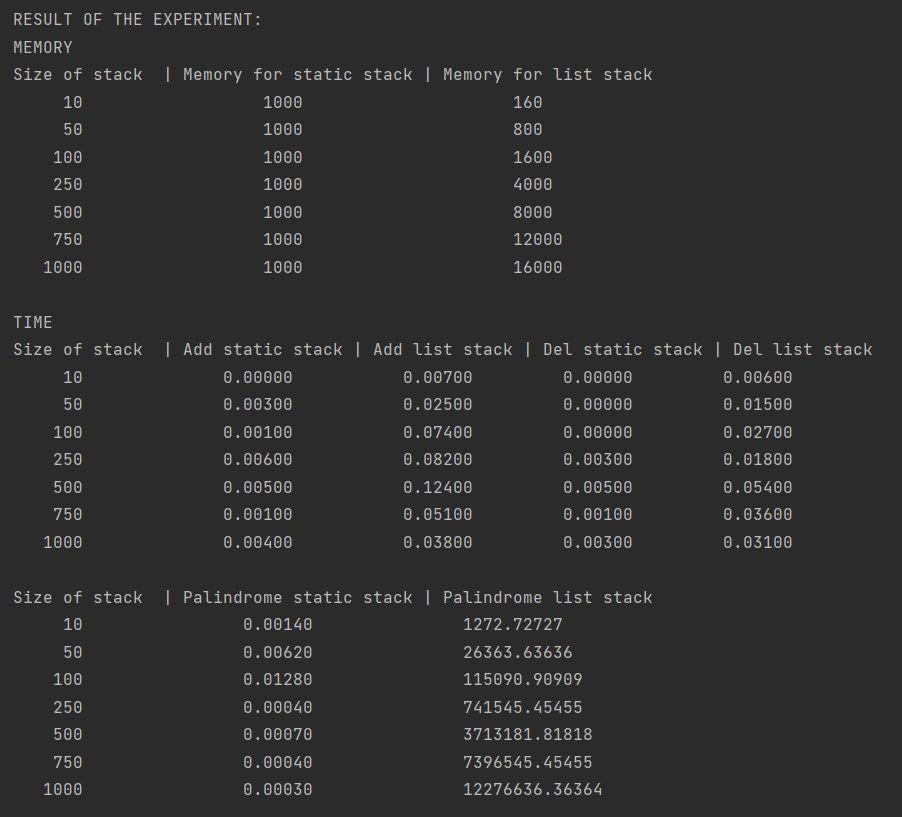
Пример вывода свободных адресов:



Пример вывода результатов исследования:



**Результаты исследования занимаемой стеками памяти и времени выполнения операций при разных размерах стеков**



**Контрольные вопросы**

1. Что такое стек?

Стек – это последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны – с его вершины. Стек функционирует по принципу: LIFO - последним пришел – первым ушел.

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

Если хранить стек как список, то память выделяется в куче. Если хранить как массив — либо в куче, либо на стеке (зависит от того, динамически или статический массив используется). Для каждого элемента стека, который хранится как список, выделяется на 4 или 8 байт (если брать современные ПК) больше, чем для элемента стека, который хранится как массив. Данные байты использованы для хранения указателя на следующий элемент списка. (из-за этого либо 4, либо 8 байт)

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

Если хранить стек как список, то верхний элемент удаляется при помощи операции освобождения памяти для него и смещением указателя, который указывает на начало стека.

При хранении стека как массива, память удаляется при завершении программы при вызове функции free (если работа идет с динамической памятью).

4. Что происходит с элементами стека при его просмотре?

Элементы стека удаляются, так как каждый раз достается верхний элемент стека, чтобы посмотреть следующий.

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

Стек эффективнее реализовать с помощью массива, так как он выигрывает в количестве занимаемой памяти (роль играет процент заполнения) и во времени обработки стека.

**Вывод**

Для реализации стека выгодно использовать массив, так как он занимает меньше памяти (для 100 элементов в 1,6 раз, для 1000 элементов уже в 16), чем список, и работает быстрее (в 6-7 раз).

Однако при использовании массива размер стека сильно ограничен. Если массив был создан статически, то его размер не изменить, если динамически – придется постоянно перевыделять память. При использовании же списка такой проблемы не возникает.