Отчет по 4 Лабораторной работе

Предмет «Типы и структуры данных»

Вариант 25

Выполнил: Чалый Андрей

Группа: ИУ7-32Б

Тема «Стек и операции над ним»

**Техническое задание**

**Задача:**

Создать программу работы со стеком, выполняющую операции добавления, удаления элементов и вывод текущего состояния стека, выполнить особое задание – Элементами стека являются адреса памяти. При реализации массивами - их вводить, при реализации списком – брать адрес выделенной памяти под элемент. Сравнить реализации стека на статическом массиве и односвязном списке. Если стек реализован на односвязном списке, то проверить, происходит ли фрагментация памяти.

**Входные данные:**

Числа вводимые в стек представленный в виде массива, либо номера команд. Выбор реализаций.

**Выходные данные:**

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

**Функция программы:**

Реализация работы со стеком. Демонстрация распределения памяти при работе со стеком. Анализ времени и объема памяти, которое требуется на добавление и удаление элементов из стека при реализации статическим массивом и списком. Реализация специального задания.

**Аварийные ситуации:**

1. Попытка добавить элемент в заполненный стек.

Программа выводит “STACK OVERFLOW”

2. Попытка убрать элемент из пустого стека.

Программа выводит “STACK UNDERFLOW”

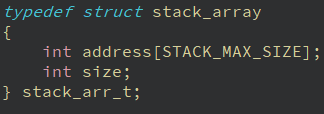
3. Попытка вывести элементы пустого стека

Программа выводит “NULL STACK”

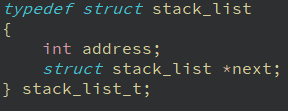
**Используемые структуры**

Для реализации структур в программе используется следующая константа: это максимальный размер стека.#define STACK\_MAX\_SIZE 100000

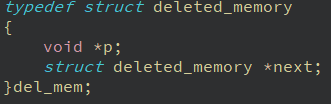
Реализация стека на статическом массиве:



Реализация стека на односвязном списке:

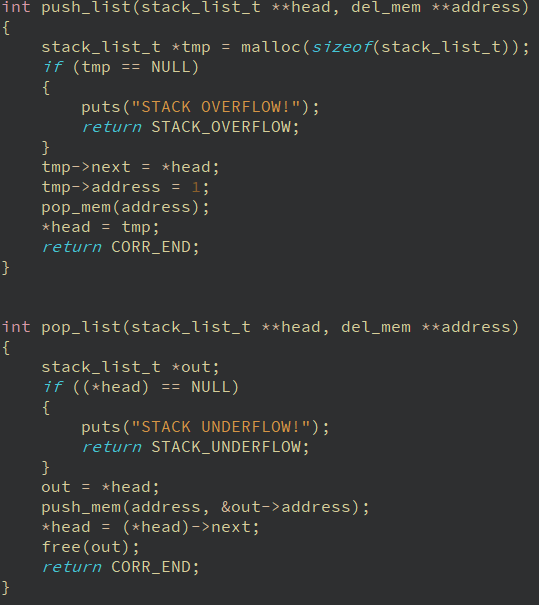


Приведенная ниже структура используется для хранения освобожденной памяти (памяти, в которой ранее хранились переменные из односвязного списка, но были удалены):

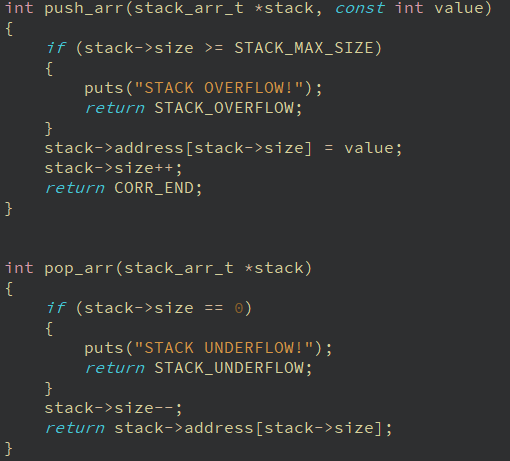


**Алгоритмы**

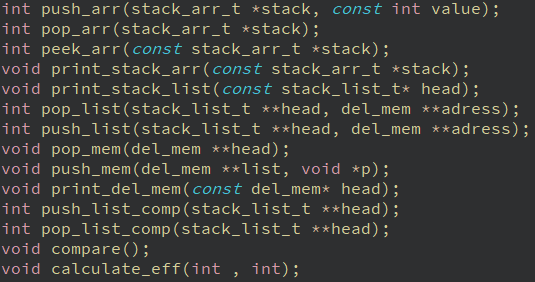
Реализация на односвязном списке.



Реализация на статическом массиве.



**Функции**



**Тесты**

Попытка добавить элемент в заполненный стек: Программа выводит “STACK OVERFLOW”

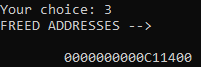
Попытка убрать элемент из пустого стека: Программа выводит “STACK UNDERFLOW”

Попытка вывести элементы пустого стека: Программа выводит “NULL STACK”

Добавляем элемент в стек (односвязный список)

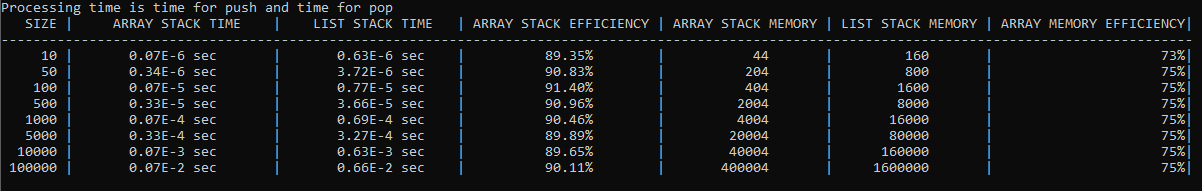


Удаляем элемент из стека, затем выводим освобожденные адреса



**Эффективность**

Эффективность того или иного способа задания стека проверяется на выполнении при разных размерах операций добавления и удаления элемента. Производится сравнение по времени и по памяти.



Мы видим, что работа с односвязным списком происходит намного медленнее, стек реализованный на статическом массиве эффективнее в 9-10 раз. По памяти, стек реализованный на статическом массиве, тоже намного эффективнее, ведь для каждого элемента стека в односвязном списке мы храним еще и указатель.

**Вывод**

Можно сделать вывод, что использование статического массива в целом более целесообразно. Этот способ быстрее и тратит меньше памяти. Однако в некоторых случаях можно использовать и односвязный список, он менее ограничен в размере, чем массив. Также в ходе лабораторной было выявлено, что при использовании односвязного списка не происходит фрагментации памяти (элементы стека располагаются в памяти друг за другом)

**Ответы на вопросы**

1. Что такое стек?

Стек – последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны. Функционирует по Last In First Out (LIFO).

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

При реализации стека списком, память выделяется динамически по мере добавления новых элементов; число элементов в стеке ограничено только количеством доступной ОП.

При реализации стека массивом, выделяется фиксированный участок памяти; в стеке не может быть больше заданного числа элементов. Добавление нового элемента происходит путём смещения индекса вершины.

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

При реализации списком память из-под элемента освобождается после его удаления.

При реализации массивом память из-под элемента не освобождается, происходит лишь изменение значения индекса вершины.

4. Что происходит с элементами стека при его просмотре?

В общем случае доступ есть только к вершине стека; при просмотре она удаляется из стека, а указатель смещается далее. Для отображения состояния стека требуется последовательно проходить по всем его элементам, не «снимая» их.

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

Реализация стека массивом даёт огромный выигрыш во времени, поскольку не нужно каждый раз заново выделять и освобождать память. Тем не менее, в этом случае количество элементов в стеке жёстко ограничено – возможно либо переполнение стека, либо постоянный «излишек» памяти, отведённой под него.

Способ реализации напрямую зависит от условий решаемой задачи – нужно знать примерное число элементов, которые могут храниться в стеке; можно ли пренебрегать переполнением; ограничен ли объём памяти.