**Отчет по лабораторной работе 4**

**По предмету «Типы и структуры данных»**

Студент ИУ7-32Б

Герасименко Екатерина

**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.

**Условие задачи:** При реализации стека массивом располагать два стека в одном массиве. Один стек располагается в начале массива и растет к концу, а другой располагается в конце массива и растет к началу. Заполнять и освобождать стеки произвольным образом с экрана. Элементами стека являются вещественные числа. Списком реализовать один стек.

**Входные данные:**

Элемент стека.

**Выходные данные:**

Стек

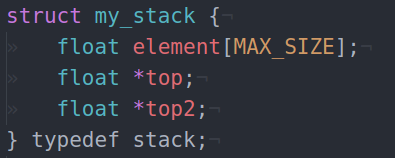
**Возможные ошибки:**

1. Выход за границы массива
2. Переполнение стека
3. Ввод некорректных данных
4. Попытка удаления элемента из пустого стека
5. Попытка вывести пустой стек

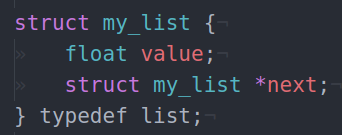
**Обращение к программе** осуществляется через консоль.

**Структуры данных:**

Реализация стека-массива:

****

Структура узла для списка:

****

**Функции:**

Взаимодействие со стеком-списком:

void insert\_list(list \*\*head)

int pop\_list(list \*\*head, void \*\*\*ptr, int \*size)

int print\_list(list \*head)

void print\_freeded(void \*\*ptr, int size)

Взаимодействие со стеком-массивом:

void init(stack \*stk, int size)

int insert(stack \*stk, int param)

float pop(stack \*stk, int param, int size)

void print(stack \*stk, int size)

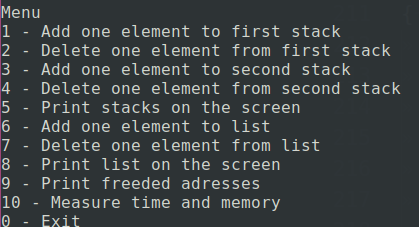
Анализ времени и памяти:

unsigned long long tick(void)

void time\_memory()

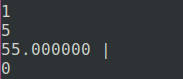
**Интерфейс:**

Главное меню:

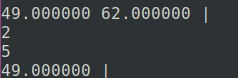


Ввод и распечатка элемента:

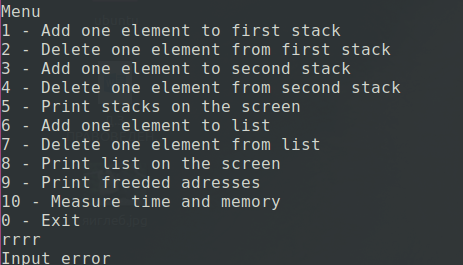
(При выборе пункта добавления числа генерируются случайным образом)

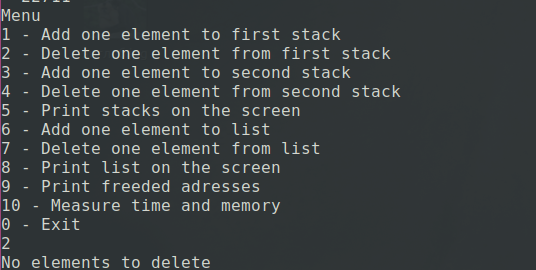


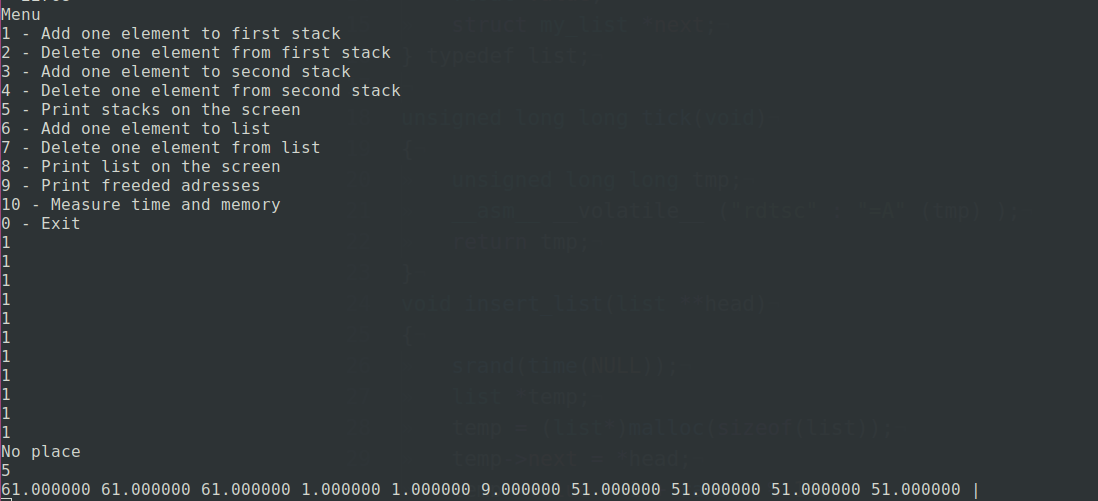
Удаление элемента:

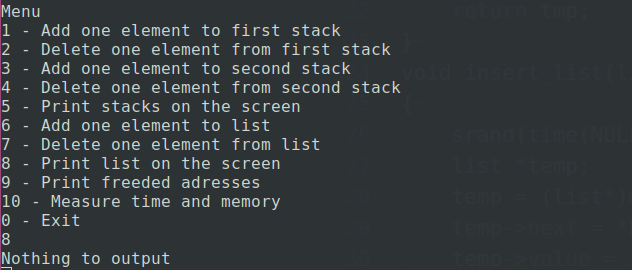


**Тесты:**

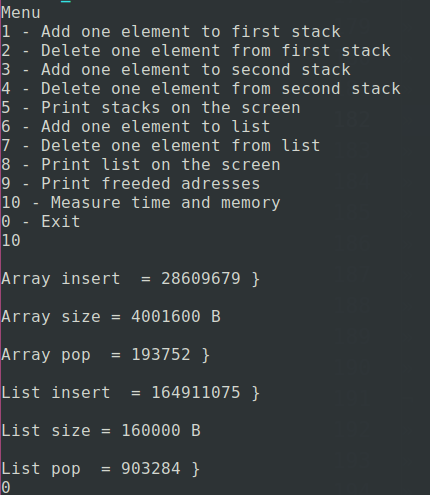








**Анализ эффективности (по памяти и времени):**

****

Работа с массивом происходит быстрее чем со списком.

**Контрольные вопросы:**

1. **Что такое стек?**

Стек – последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с одной стороны. Функционирует по принципу «последний зашел – первый вышел».

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.

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

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

При реализации списком память из-под элемента освобождается при его удалении.

При реализации массивом память из-под элемента не освобождается, происходит лишь изменение значения указателя на последний элемент.

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

Мы храним только указатель на последний элемент, чтобы пройтись по стеку нужно обойти все элементы стека по указателям на предыдущий элемент.

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

Реализация стека массивом дает 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.

При реализации стека списком возникает меньше забот с памятью.

**Вывод:**

Реализация стека массивом позволяет выиграть по времени, но проигрывают по памяти спискам(в моей реализации). Но для удаления элементов из списка требуется больше времени, чем для удаления из массива. Так что выбор метода реализации зависит от требований и того, что важнее в данной реализации : память или время.