**Лабораторная работа №4**

**Условие**

Вариант 7

Реализовать операции добавления, удаления элемента и вывода состояния стека, который представлен в виде массива и односвязного линейного списка. Оценить преимущества и недостатки каждой реализации. При реализации стека массивом располагать 2 стека в одном массиве. Элементы стека – вещественные числа.

**Наименование**

Программа для реализации стека и сравнения двух способов реализации стека – массивом и списком.

**Область применения**

Выполнение вычислений математических выражений, в частности вычисление многоступенчатых выражений.

**Сроки исполнения**

2 недели, начиная с 29.10.2018

**Основание для разработки**

Заказчик – Силантьева Александра Васильевна

Исполнитель – Ильясов Идрис Магомет-Салиевич, группа ИУ7-33Б

Основание для разработки – лабораторная работа №4

**Назначение разработки**

Общая концепция системы – консольное приложение.

Описание функционала системы:

Ввод и вывод производится пользователем вручную через консоль с помощью меню.

В меню пользователь может добавить элемент в стек, удалить элемент из стека, провести сравнение реализаций. При работе с программой стеки выводятся автоматически.

Добавление и удаление элементов происходит по одному.

Изначально стеки пустые. Максимальный размер стека – 10 элементов. Максимальный размер стека изменяется вручную в файлах.

**Требования к программе**

Программа написана полностью на языке Си.

При неправильном вводе пользователем какого-либо значения программа завершает свою работу, выдавая сообщение об ошибке.

**Структуры данных и функции**

Стек-массив представляет собой структуру:

**struct stack**

**{**

**float data[STACKSIZE \* 2];**

**int pos\_l;**

**int pos\_r;**

**};**

pos\_l – вершина первого стека, pos\_r – вершина второго стека.

Стек-список представляет собой структуру:

**typedef struct StackNode**

**{**

**float data;**

**struct StackNode \*next;**

**} StackNode;**

**Модули и тесты**

Программа не нуждается в спецификации модулей и модульных тестах.

**Функциональные тесты**

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Вывод** | **Класс эквивалентности** |
| Меню: y | Сообщение об ошибке, завершение программы | Некорректный ввод в меню |
| Меню: 3 | Успешный ввод | Корректный ввод в меню |
| Ввод элемента стека при добавлении: а | Сообщение об ошибке, завершение программы | Некорректный ввод при добавлении элемента в стек |
| Ввод элемента стека при добавлении: 3 | Успешный ввод | Корректный ввод при добавлении элемента в стек |

**Алгоритм**

Ввод: choice

Вывод: –

Считать choice

Пока TRUE

Если choice = 1

Добавить элемент в левый стек-массив

Если choice = 2

Добавить элемент в правый стек-массив

Если choice = 3

Удалить элемент из левого стек-массива

Если choice = 4

Удалить элемент из правого стек-массива

Если choice = 5

Добавить элемент в стек-список

Если choice = 6

Удалить элемент из стек-списка

Если choice = 0

Выйти

Иначе

Вывести сообщение об ошибке и выйти

Все если

Все пока

**Выводы**

В ходе выполнения данной лабораторной работы я научился реализовывать стек двумя способами – массивом и списком. Также была сравнена эффективность этих способов по памяти и времени, затрачиваемому на добавление и удаление элементов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Количество элементов | Массив, добавление | Массив, удаление | Массив, память | Список, добавление | Список, удаление | Список, память |
| 10 | 0.003ms | 0.002ms | 880b | 0.020ms | 0.018ms | 160b |
| 100 | 0.02ms | 0.013ms | 8800b | 0.025ms | 0.021ms | 1600b |
| 1000 | 0.12ms | 0.097ms | 88000b | 0.063ms | 0.056ms | 16000b |

Также стоит отметить, что работать со стеком-массивом эффективнее, чем работать со стеком-списком. По результатам тестирования видно, что при добавлении и удалении элемента массивы значительно быстрее списков из-за отсутствия необходимости выделения/освобождения памяти под каждый элемент.

При реализации стека списком элементы располагаются в определенных участках памяти. Также стоит отметить, что фрагментации памяти не наблюдается.

**Контрольные вопросы**

1. Что такое стек?

Стек – это последовательный список, включение и исключение элемента в котором происходит только с одной стороны по принципу “Last-in-First-out” (LIFO).

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

При хранении стека в виде массива под него выделяется определенный объем памяти. При хранении стека в виде списка объем памяти ограничен объемом доступной оперативной памяти, но может быть ограничен искусственно.

Память под массив выделяется len\*sizeof(элемент). Память для списка выделяется не только под элементы самого списка, но также и под указатели.

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

При реализации стека массивом память освобождается в конце области видимости массива при уничтожении соответствующей переменной. При реализации стека списком память освобождается при каждом удалении элемента.

4. Что происходит с элементами стека при его просмотре?

При просмотре стека он очищается.

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

Эффективнее реализовывать стек массивом, потому что добавление/удаление осуществляются просто по индексу, а память выделяется один раз (несколько раз, если стек расширяемый).