

	<p>Министерство образования и науки 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)</p>
---	--

ФАКУЛЬТЕТ _____ Информатика и системы управления (ИУ) _____

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии (ИУ7) _____

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4 **«Работа со стеком»**

Студент, группа

Сабуров С., ИУ7-33Б

2020 г.

1. Цель работы.

Цель работы: Обрести навыки работы со стеком. Реализовать стек как в виде массива, так и в виде односвязного списка. Реализовать операции для работы со стеком. Выяснить принцип выделения и освобождения памяти.

Описание условия задачи:

Проверить правильность расстановки скобок трёх видов(круглые, квадратные, фигурные) в выражении.

2. Техническое задание.

Требования к программе.

- 1) Предоставление списка действий,производимых программой.
- 2) Наличие пояснений при выводе результата.
- 3) проверка правильности расстановки скобок в выражении

Тип входных данных:

- 1) Цифра, которая соответствует определенному действию.
- 2) Строка, содержащая искомое выражение.

Тип выходных данных:

- 1) Результат выполнения индивидуального задания.
- 2) Количественная характеристика обработки стека.

Способ обращения к программе: Сперва необходимо произвести сборку проекта, прописав «make».Запуск программы осуществляется в терминале ОС linux при помощи строки ./app.exe , далее происходит ввод числа , соответствующего определенному действию. Название программы — app.exe

Аварийные ситуации.

1. Некорректный ввод выбора структуры стека:

```
Choose struct:
1 - array
2 - list
3 - Exit
4
ERROR INCORRECT CHOICE
```

2. Некорректный выбор операции над стеком:

```
1 - solve problem
2 - add element
3 - delete element
4 - show stack
0 - exit
q
ERROR IN CHOICE
```

Структуры данных.

```
typedef struct
```

```
{
```

```
    char array[MAX_STACK_SIZE];
```

```
    int len;
```

```
}Stack_arr; - Структура стека в виде массива
```

```
typedef struct node
```

```
{
```

```
    char bracket;
```

```
    struct node *next;
```

```
    int ind;
```

```
}Stack_list; - Структура стека в виде линейного односвязного списка
```

```
typedef struct array
{
    size_t arr[MAX_STACK_SIZE];
    int ind;
}free_sp; - Структура массива свободных областей
```

Алгоритм (полный алгоритм работы программы).

- 1) Ввод команды из меню.
- 2) Предлагается выбрать один из пунктов меню, пока не будет введён 0.

Пункты меню:

Choose struct:

- 1 — **array** — выбрать массив в качестве структуры для реализации стека
- 2 — **list** — выбрать линейный односвязный список в качестве структуры для реализации стека
- 3 — **Exit** — Выход

Подпункты меню:

- 1 - **solve problem** — Решение индивидуального задания
- 2 - **add elements** — добавление элементов в стек
- 3 - **delete element** — удалить 1 элемент
- 4 - **show stack** — показать текущее состояние стека
- 5 - **show free space** — распечатать массив свободных областей (для списка)
- 6 - **delete stack** — очистить стек
- 0 — **exit** — возврат к выбору структуры

Тесты.

№	Тест	Ввод	Результат
1	Некорректный ввод действия при выборе структуры	4	ERROR INCORRECT CHOICE
2	Некорректный ввод действия при выборе операции над стеком	A	ERROR IN CHOICE
3	Превышение лимита при вводе элементов	10 элементов (лимит = 9)	ERROR OVERFLOW
4	Удаление элемента	Стек(1,2,3)	Стек(1, 2) Время = время
5	Удаление элемента в пустом стеке	Стек()	ERROR: STACK IS EMPTY. NOTHING TO DELETE
6	Добавление элемента в стек	Стек(1,2,3) Элемент = 4	Стек(1,2,3, 4) Время = время
7	Добавление элемента в полный стек	Стек(1, 2, 3) Элемент = 4	ERROR: STACK OVERFLOW
8	Индивидуальное задание Корректная расстановка скобок	(2 + 3)	Время = время
9	Индивидуальное задание Некорректная расстановка скобок	2 +) 3 (ERROR: Incorrect brackets

Оценка эффективности:

Измерение эффективности умножения производится в тактах процессора.

ВРЕМЯ

Добавления элемента

Количество элементов	Список	Массив
10	2352	796
100	19032	2544
1000	239156	21302

Удаление элемента

Количество элементов	Список	Массив
10	3192	436
100	9604	1074
1000	65674	7104

ОБЪЁМ ЗАНИМАЕМОЙ ПАМЯТИ(в байтах):

Количество элементов	Список	Массив
10	174	14
100	1704	104
1000	17004	1004

Вывод:

В ходе лабораторной работы было выявлено, что стек, реализованный в виде массива имеет превосходство над стеком, реализованным в виде линейного односвязного списка как по времени(~3 раза), так и по памяти(~12 раз). Таким

образом, можно сделать вывод, что при реализации такой структуры данных как стек, следует выбирать массив.

Ответы на вопросы

1. Что такое стек?

Стек — структура данных, в который можно обрабатывать только верний элемент. Операции производятся согласно правилу: последним зашел — первым вышел.

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

При реализации стека в виде списка, память выделяется в куче. При реализации стека в виде массива, память выделяется либо в куче (динамический массив), либо в стеке (статический). Под список выделяется больше памяти, поскольку, помимо полезной информации, следует хранить указатель на следующий элемент.

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

При реализации в виде списка, память освобождается посредством освобождения памяти из под элемента и смещения указателя на следующий элемент. В случае же массива смещается лишь указатель на вершину.

4. Что происходит с элементами стека при его просмотре?

Элементы стека уничтожаются, так как каждый раз достигается его верхушка.

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

Эффективнее реализовывать стек в виде массива. Он выигрывает как по времени, так и по памяти.