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)

ФАКУЛЬТЕТ Информатика и системы управления (ИУ)

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ7)

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4
«РАБОТА СО СТЕКОМ»

Студент, группа **Буланый К., ИУ7-36Б**

2020 г.

Описание условия задачи

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

1) массивом;
2) 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.
При реализации стека массивом располагать два стека в одном массиве. Один стек располагается в начале массива и растёт к концу, а другой располагается в конце массива и растёт к началу. Заполнять и освобождать стеки произвольным образом с экрана. Элементами стека являются вещественные числа. Списком реализовать один стек.

Техническое задание

Входные данные

1. Целое число, представляющее собой номер команды: от 0 до 2.
2. Командно-зависимые данные:
   - операция со стеком;
   - количество элементов в массиве;
   - элементы массива.

Выходные данные

1. Результат операции.
2. Время выполнения операции.

Функция программы

Программа выполняет ряд функций, указанный при запуске.
Она позволяет:

1. Выбрать тип стека.
2. Добавить элемент в стек.
3. Удалить элемент из стека.
4. Очистить стек.
5. Вывести текущее содержимое стека.

Обращение к программе:

Запускается из терминала.

Аварийные ситуации

1. Некорректный ввод номера команды. На входе: число, не указанное в меню. На выходе: сообщение «Unknown command.».
2. Некорректный ввод количества элементов массива. На входе: неположительное или нецелое число или не число. На выходе: сообщение «Wrong size.».
3. Некорректный ввод максимального количества элементов стека списком. На входе: неположительное или нецелое число или не число. На выходе: сообщение «Wrong max size.».
4. Некорректный элемент стека. На входе: не число. На выходе: сообщение «Wrong number.».

Структуры данных

Реализация на основе массива:

```
typedef struct arrstack
{
	double *arr;
	double *top1;
	double *end;
	double *top2;
} arrstack_t;
```

Поля структуры:

1. double *arr — указатель на начало массива/низ первого стека.
2. double *top1 — указатель на верх первого стека.
3. double *end — указатель на конец массива/низ второго стека.
4. double *top2 — указатель на верх второго стека.

Реализация на основе односвязного списка:

```
struct stacknode
{
	double num;
	stacknode_t *next;
};
```

Поля структуры:

1. double num — содержимое элемента стека.
2. stacknode_t *next — следующий элемент стека.

Алгоритм

1. Пользователь вводит номер команды из меню.

2. Пока пользователь не введёт 0, ему будет предложен выбор реализации стека.
3. После выбора типа стека пользователю будут предложены действия со стеком, пока он не введёт 0.
4. Пользователь может:
   - удалить элемент из стека;
   - добавить элемент в стек;
   - очистить стек;
   - задать размер стека;
   - вывести стек.

Тесты

	Тест	Ввод	Результат
1	Неверная команда	13	Wrong command.
2	Неверный размер стека/массива	-1	Wrong [max] size.
3	Неверный элемент стека	A	Wrong number.
4	Добавление в полный стек массивом.	(команда)	Stacks are full.
5	Добавление в полный стек списком	(команда)	Can't push. Stack is full.
6	Извлечение из пустого стека списком	(команда)	Stack is empty.
7	Извлечение из пустого первого стека массивом	(команда)	Couldn't pop a number. First stack is empty.
8	Извлечение из пустого второго стека массивом	(команда)	Couldn't pop a number. Second stack is empty.
9	Попытка добавить, удалить элемент, очистить или вывести несозданный стек массивом.	(команда)	A stack wasn't created.
10	Верное добавление в стек	1	Time of pushing: X. Done.
11	Верное извлечение из стека	(команда)	Number: X. Time of popping a number: Y
12	Вывод стека	(команда)	(стек)
13	Очищение стека	(команда)	Time of emptying: X . Done.

Оценка эффективности

Время добавления элемента (в тактах)

Массив	Список
112	4672

Время извлечения элемента (в тактах)

Массив	Список
100	4136

Время очищения стека (в тактах)

Количество	Массив	Список
10	104	7108
100	110	60152

Объем стека (в байтах)

Количество	Массив	Список
10	112	160
100	832	1600

Контрольные вопросы

1. Что такое стек?

Стек – структура данных, в которой можно обрабатывать только последний добавленный элемент. На стек действует правило LIFO — последним пришел, первым вышел.

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

При хранении стека с помощью списка, то память всегда выделяется в куче. При хранении с помощью массива, память выделяется либо в куче, либо на стеке (в зависимости от того, динамический массив или статический).

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

При хранении стека связанным списком, верхний элемент удаляется освобождением памяти для него и смещением указателя, указывающего на начало стека. При удалении из стека, реализованного массивом, смещается лишь указатель на вершину стека.

4. Что происходит с элементами стека при его просмотре?

Элементы стека уничтожаются, так как каждый раз достаётся верхний элемент стека.

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

Реализовывать стек эффективнее с помощью массива. Он выигрывает как во времени обработки, так и в количестве занимаемой памяти. Однако в случае реализации списком, память для списка ограничена лишь размером оперативной памяти, в то время как стек массивом нужно увеличивать или создавать заново.

Вывод

Стек, реализованный односвязным списком очень серьёзно проигрывает списку, реализованным массивом, по времени обработки, а также, но не так значительно, и по объёму занимаемой памяти. В то же время, стек, реализованный списком, не нужно создавать или расширять, так как элементы стека не зависят друга от друга физически.