Министерство науки и высшего образования 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)

Лабораторная Работа №2 «Работа со стеком»

Вариант №5

Студент **Шахнович Дмитрий Сергеевич**

Группа **ИУ7-22Б**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент **Шахнович Д.С.**

Оценка _____

2023 г.

Описание условия задачи

создать программу работы со стеком, выполняющую операции добавление, удаления элементов и вывод текущего состояния стека. Реализовать стек а) статическим массивом (дополнительно можно реализовать динамическим массивом); б) 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.

Ввести арифметическое выражение типа: число|знак| ... число|знак| число. Вычислить значение выражения. (Приоритетность операций необязательна)

Техническое задание

Исходные данные:

Файл для стандартного стека: В файле на каждой строке находятся числа.

Файл для стека операций: В файле в одной строке задано выражение в формате число, знак, число… разделенные пробелом.

В меню действия заданы числами 0-6, которые включают работу с текущим стеком и выполнение арифметического выражения. Все действия должны быть возможны как со статическим стеком, так и стеком-списком.

Выходные данные:

Программа может выдавать текущее состояние стека, а также высчитывать арифметические выражения в заявленном формате.

Описание задания:

Чтение, обработка, вывод на со стеком, а также выполнение арифметических операций с его помощью.

Способы обращения к программе:

Запуск программы через терминал, затем управление программой с помощью меню. Пункты меню:

1 — Считать стек из файла.

2 — Добавить элемент в стек.

3 — Удалить элемент из стека.

4 — Посчитать арифметическое выражение.

5 — Сравнить выполнение арифметических выражений на списке-стеке и статическом стеке.

6 — Вывести текущий стек.

0 — Выход.

Аварийные ситуации:

1) Ввод несуществующей команды в меню;
   Сообщение: «Ошибка: Некорректная команда.»
2) Ошибка ввода/вывода;
   Сообщение: «Ошибка функций ввода/вывода.»
3) Удаление элемента из пустого стека.;
   Сообщение: «Ошибка: Стек пустой.»
4) Переполнение статического стека;
   Сообщение: «Ошибка: Стек переполнен.»
5) Ввод некорректного имени файла;
   Сообщение: «Ошибка: Неправильное имя файла.»
6) Неудачная попытка выделения памяти;

Сообщение: «Ошибка выделения памяти.»

7) Ввод литералов или чисел вне запрашиваемого диапазона;

   Сообщение: «Ошибка: Некорректный формат ввода.»

8) Ввод арифметического выражения с делением на ноль;

   Сообщение: «Ошибка: Деление на ноль.»

9) Неудачная попытка работы с файлом;

   Сообщение: «Ошибка файлового ввода/вывода.»

Описание структур данных

```
/// @brief Узел стека ввиде односвязного списка
struct stack_node_t
{
   int data; /// Значение узла
   stack_node_t *next; /// Указатель на следующий(более близкий к начальному) элемент стека.
};
```

Структура стека в виде односвязного списка+.

```
/// @brief Реализация стека с помощью статического массива
struct static_stack_t
{
   int *head_ptr; /// Указатель на голову стека
   int *end_ptr; /// Указатель на конец статического массива
   int arr[STATIC_STACK_SIZE]; /// Статический массив
};
```

Структура стека в виде статического массива

Описание алгоритма

1. Вывести пользователю меню и ожидать ввода номера команды;
2. В случае работы со стеком вывести запрос на ввод файла, элемента и т. д.;
3. В случае вычисления арифметического выражения запросить файл с выражением или ручной ввод выражения;
4. В случае сравнения методов хранения запросить файл с выражением;

При возникновении ошибок или завершения пункта вернуться к выбору пункта меню.

Тестовые данные

Позитивные тесты			
№	Описание	Вход	Выход
1	Добавить в пустой стек один элемент	Стек: пустой Элемент 3	Стек: 3
2	Удалить из стека с одним элементом один элемент	Стек: 3	Стек: пустой
3	Заполнить пустой стек 3-мя элементами	Стек: пустой Элементый 1, 2, 3	Стек:3 2 1
4	Удалить из стека с 3-мя элементами один элемент	Стек:3 2 1	Стек:2 1
5	Ввести стек из файла	./data/standard/ data_3.txt	Стек: -74 -64 40
6	Выполнить выражения с двумя элементами	1 + 2 - 3	2
7	Выполнить выражение с умножением	2 + 2 * 4	10
8	Выполнить выражение с делением	2 + 10 / 2	7
9	Выполнить комплексное выражение	2 + 2 * 3 / 3 * 2	6
10	Выполнить выражения, заданное файлом	./data/operation/ data_100.txt	-9757677

Негативные тесты			
1	Некорректный формат файла со стеком	1 2 ad 2	Ошибка: Некорректный формат ввода.
2	Ввести неверный код в меню	12	Ошибка: Некорректная команда.
3	При запросе числа ввести литерал	sda	Ошибка: Некорректный формат ввода.
4	Ввести число вне указанного диапазона	(1 — Консольный, 0 — файловый) - 3	Ошибка: Некорректный формат ввода.
5	Ввести несуществующее название файла	{Не сущ. файл}	Ошибка: Неправильное имя файла.
6	Ввести выражение с делением на ноль	0 / 10	Ошибка: Деление на ноль.
7	Задать выражение, оканчивающееся на оператор	1 + 2 *	Ошибка: Некорректный формат ввода.
8	Удалить из пустого стека	Стек: пустой	Ошибка: Стек пустой.
9	Добавить элемент в полный стек	Стек: полный	Ошибка: Стек переполнен.

Замеры выполнения операций

Замеры операций проводились следующим образом: с помощью скрипта создавались файлы разных размеров со случайными выражениями. Затем вычисления для каждого из данных файлов проводились 1000 раз, при этом замерялось только время вычисления, время чтения файла, создания массивов и т. д. не учитывалось. Размер стека указывается по количеству операций в файле.

Время выполнения вычислений

Размер стека	Время в статическом стеке, мкс	Время в списке-стеке, мкс	Отношение времени списка к статическому
500	5	8	1.6
1000	13	18	1.4
2000	19	40	2.1
5000	63	108	1.7
7500	101	163	1.6
10000	155	228	1.5

Объемы памяти

Размер стека	Объем статического стека, байт	Объем списка-стека,байт	Отношения объема списка к статическому
500	4036	16032	~4
1000	8036	32032	~4
2000	16036	64032	~4
5000	40036	160032	~4
7500	60036	240032	~4
10000	80036	320032	~4

Как видно из таблиц использование стека в виде односвязного стека вместо статического массива увеличивает объем памяти в 4 раза, и при этом замедляет выполнение задачи примерно 1.7 раз.

Ответы на вопросы

1. *Что такое стек?*

   Стек — последовательный список, вариантной длины, в котором включение и исключение элементов происходит с одной стороны - с его головы. Основной принцип — LIFO — last in, first out

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

   В зависимости от реализации под стек может выделяться разное количество памяти, например, в случае реализации стека через статический массив память выделяется в стеке и задается константным значением. В случае реализации в виде односвязного списка память выделяется в куче блоками пар данные-указатель на следующий элемент списка. При этом при добавлении элемента выделяется память под новый блок, а при удалении — освобождается.

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

   При реализации статическим массивом память не освобождается. При реализации динамически расширяемым массивом память может освободиться при перевыделении массива. В случае списка память из под блока сразу освобождается при удалении.

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

   При реализации классического стека невозможно просмотреть элемент не удаляя его, поэтому при просмотре элемента он удаляется из стека.

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

   В зависимости от целей стек можно реализовывать по-разному. Реализовывать его как статический массив выгодно при известном

небольшом максимуме количество элементов и если в данном случае важно скорость. Наиболее быстрой в написании и удобной является реализация через списки. Реализация через динамически расширяемые массивы находится где-то посередине между предыдущими, так как обладает частью их преимуществ и недостатков.

Выводы

Во многих компьютерных системах и алгоритмах используется тип данных стек, при этом как у любого другого абстрактного типа данных существует множество его реализаций. Задача программиста при конкретной задаче выбрать наиболее оптимальную реализацию, зависящую от специфики области, и использовать её.