|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 №4**

**По дисциплине «Типы и структуры данных»  
  
Тема: «Стек»**

**Вариант 9**

* + 1. Выполнил: студент группы ИУ7-36(Б)  
       Ковель Александр Денисович
    2. Сдал:

**Москва, 2021**

**Цель работы**

Реализовать операции работы со стеком, который представлен в виде массива (статического или динамического) и в виде односвязного списка, оценить преимущества и недостатки каждой реализации, получить представление о механизмах выделения и освобождения памяти при работе с динамическими структурами данных.

**Задание**

Разработать программу работы со стеком, реализующую операции добавления и удаления элементов из стека и отображения текущего состояния стека. Реализовать стек:

а) массивом;

б) списком.

Все стандартные операции со стеком должны быть оформлены отдельными

подпрограммами. В случае реализации стека в виде списка при отображении

текущего состояния стека предусмотреть возможность просмотра адресов элементов стека и создания дополнительного собственного списка свободных областей (адресов освобождаемой памяти при удалении элемента, который можно реализовать как списком, так и массивом) с выводом его на экран. Список свободных областей необходим для того, чтобы проследить, каким образом происходит выделение памяти менеджером памяти при запросах на нее и убедиться в возникновении или отсутствии фрагментации памяти.

**Техническое задание**

**Общая концепция системы**

Программа предназначена для перевода выражения в постфиксную форму.

**Требования к функциональным характеристикам**

Программа должна выполнять следующие функции:

* Добавить элемента в стек
* Удалить элемент из стека
* Вывести постфиксную запись
* Вывести время обработки
* Отобразить пояснения при выводе результата

**Вход**

На вход программа получает символы, которые добавляются в стек.

**Выход**

Постфиксная запись введенного выражения.

**Аварийные ситуации**

1. Удаления пустого стека

Программа выведет сообщение “Стек пуст”.

2. Вывод списка пустого

Программа выведет сообщение “Стек пуст”.

3. Переполнение стека

Программа выведет сообщение “Ничего не добавлено”.

**Способ обращения к программе**

Программа представляет собой файл app.exe. Запускается в консоли. для запуска достаточно команды ./app.exe. Если файл отсутствует, можно собрать его с помощью утилиты make.

**Описание структуры данных**

typedef struct stack

{

    int data[MAXSIZE]; // массив

    int top; // Количество элементов

}stack;

struct list\_stack

{

    int data; // элемент

    list\_stack \*next; // ссылка на следующий элемент

};

**Описание алгоритма**

Пока не выбрана команда выход:

1. Добавить элемент
   1. Если стек не переполнен, добавить
   2. Удалить из массива удаленных ячеек, заполненный адрес (список)
2. Удалить элемент
   1. Если стек не пуст
   2. Добавить его в массив удаленных (если это список)
3. Вывод стека (если это список)
   1. Вывод самого списка
   2. Адреса удаленных ячеек стека
4. Вывод постфиксной записи
   1. Если не скобка и не знак записываем в строку
   2. Если скобка открывающая, то в стек
   3. Если операция, то просматриваем приоритет
5. Выход

**Тесты**

|  |  |  |
| --- | --- | --- |
| Пункт | Ввод | Выход |
| 1 | 1 |  |
| 2 |  |  |
| 2 |  | Стек пуст |
| 1 | 1 |  |
| 1 | + |  |
| 1 | 1 |  |
| 3 |  | 1 1 + |
| 4 |  | 1  адрес  +  адрес  1  Адрес  Адреса удаленных элементов |

**Оценка эффективности**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество элементов | Массив, мкс | Массив, память, байт | Список, мкс | Список, байт |
| 10 | 0.230000 | 14 | 1.000000 | 50 |
| 50 | 0.950000 | 54 | 4.260000 | 250 |
| 100 | 1.820000 | 104 | 6.740000 | 500 |

**Выводы по проделанной работе**

В ходе проделанной работы была создана программа для работы со стеком. Реализован перевод стека из инфиксной формы в постфиксную.

Стек эффективнее реализовать массивом, так как при добавлении в список, нужно выделять память. Массив эффективнее по памяти и по времени.

**Ответы на вопросы**

1. Что такое стек?

Стек – это последовательный список с переменной длиной, в котором добавление и удаление элементов происходит только с вершины стека.

2. Каким образом и сколько памяти выделяется под хранение стека при

различной его реализации?

При реализации стека массивом, память выделяется только под элемент массива. При реализации стека списком, память выделяется не только под сам элемент, но и под указатель на предыдущий элемент стека.

3. Каким образом освобождается память при удалении элемента стека при

различной реализации стека?

При реализации стека с помощью списка, освобождается память из-под элемента списка. При реализации списка массивом, меняется количество элементов массива. Массив очищается в конце программы.

4. Что происходит с элементами стека при его просмотре?

Элементы стека удаляются при его просмотре.

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

Реализовывать эффективнее массивом, если заранее известно количество элементов. Зависеть может от способа хранения массива. Если массив статический, то такая реализации будет быстрее стека в разы.