|  |  |
| --- | --- |
| 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)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **ПРОГРАММНАЯ ИНЖЕНЕРИЯ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04** Программная инженерия

**Отчет**

|  |  |
| --- | --- |
| **По лабораторной работе №** | 4 |

**Название:**

Работа со стеком

**Дисциплина:** Типы и структуры данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-36Б |  | | А.А. Жаворонкова |
|  | (Группа) | |  | (И.О. Фамилия) |
|  |  | |  |  |
| Преподаватель: | Никульшина Т. А. | |  |  |

Москва, 2022

Описание условия задачи

Создать программу работы со стеком, выполняющую операции добавления, удаления элементов и вывода текущего состояния стека. Реализовать стек: а) 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.  
Перевести выражение в постфиксную форму.

Описание ТЗ

*Описание исходных данных и результатов*

*Исходные данные:* пункт меню (см. описание задачи), элементы стека, выражение в инфиксной форме.

*Результаты:* стек в виде списка, стек в виде массива, выражение в постфиксной форме, адреса освобождаемых элементов, время работы функций push и pop.

*Описание задачи, реализуемой программой*

Меню:

Операции с массивом:  
1) Создать стек  
2) Добавить элемент в стек  
3) Удалить элемент из стека  
4) Вывести текущее состояние стека  
5) Перевести выражение в постфиксную форму

Операции со списком:  
6) Создать стек  
7) Добавить элемент в стек  
8) Удалить элемент из стека  
9) Вывести текущее состояние стека  
10) Перевести выражение в постфиксную форму  
11) Вывести адреса освобождаемых элементов  
12) Статистика работы функций  
13) Вывести меню

0) Выйти из программы

*Способ обращения к программе*

Вызов программы происходит через терминал (main.exe). Дальнейшая работа с программой выполняется при помощи меню, выводимого на экран.

*Описание возможных ошибок пользователя*

* Неверный размер стека, представленного массивом
* Добавление / удаление элемента из несозданного стека

Описание внутренних структур данных

Стек в виде массива представлен следующей структурой:

struct stack\_array

{

    char \*content;

    int top;

    int size;

};

**content –** массив символов  
**top –** индекс вершины стека  
**size –** размер стека

Стек в виде списка представлен следующими структурами:

struct node

{

    char item;

    struct node \*next;

};

struct stack\_list

{

    struct node \*top;

};

**item –** элемент стека  
**next –** указатель на следующий элемент  
**top –** указатель на вершину стека

Набор тестов с указанием проверяемого параметра

|  |  |  |
| --- | --- | --- |
| Входные данные | Выходные данные | Что проверяется |
| Некорректный размер  создаваемого стека | «Введен некорректный размер» | Создание стека |
| Количество элементов в стеке в виде массива больше размера стека | «Введено некорректное количество» | Инициализация стека |
| Добавление элемента в заполненный стек | «Стек заполнен» | Переполнение стека |
| Удаление элемента из пустого стека | «Стек пуст» | Удаление из пустого стека |
| Вывод пустого стека | «Стек пуст» | Вывод пустого стека |
| Вывод адресов освобождаемых элементов, когда ни один элемент не был удален | «Нет освобожденных участков памяти» | Вывод пустого списка адресов |

Описание алгоритма

Перевод из инфиксную в постфиксную форму.  
Считанная строка обрабатывается посимвольно. Если текущий символ является буквой или цифрой, то он сразу записывается в строку-результат. Если текущий символ – открывающаяся скобка “(”, то она кладется в стек. Если текущий символ – закрывающаяся скобка “)”, то из стека достаются все операции до открывающейся скобки и вместе с операндами записываются в строку-результат. Если текущий символ – какая-либо операция, то из стека достаются все операции, имеющие больший приоритет, чем у текущей, и также вместе с соответствующими операндами записываются в строку-результат, а текущая операция кладется в стек. Когда обработана вся строка, все операции из стека записываются в конец строки-результата.

Временная эффективность и затраты памяти

Следующие данные были получены путем измерения времени добавления / удаления элемента из стека в виде массива и стека в виде списка.

Стек на 50 элементов:

|  |  |  |  |
| --- | --- | --- | --- |
| Добавление элемента | | Удаление элемента | |
| Список | Массив | Список | Массив |
| 141 нс | 84 нс | 135 нс | 83 нс |

|  |  |
| --- | --- |
| Список | Массив |
| 808 б | 66 б |

Стек на 100 элементов:

|  |  |  |  |
| --- | --- | --- | --- |
| Добавление элемента | | Удаление элемента | |
| Список | Массив | Список | Массив |
| 141 нс | 83 нс | 140 нс | 82 нс |

|  |  |
| --- | --- |
| Список | Массив |
| 1608 б | 116 б |

Данные были получены при условии максимальной заполненности стека в виде массива. Такой стек выделяется единожды и выделенная память может оказаться существенно больше затраченной, что влечет проигрыш по памяти. Поэтому, если стек в виде массива заполняется не более, чем на 6% от выделенной памяти, то целесообразнее использовать стек в виде списка.

Вывод

Стек, реализованный в виде массива, имеет преимущество перед стеком в виде списка как по времени работы, так и по занимаемой памяти при условии заполненности стека более чем на 6%. Однако при реализации списком снижается вероятность переполнения стека, потому что стек массивом выделяется единожды, а стек списком выделяет память под каждый элемент.

Ответы на вопросы

1. Что такое стек?

Стек – это последовательный список с переменной длиной, в котором включение и исключение элементов происходит только с его вершины. Стек работает по принципу LIFO: последним пришел – первым ушел.

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

При реализации стека массивом память выделяется один раз в соответствии с заданным размером стека. При реализации списком – память выделяется под каждый новый элемент, то есть под одну структурную переменную.

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

При реализации стека массивом память освобождается только в конце программы – освобождается весь массив, а при удалении происходит сдвиг верхней границы стека. При реализации списком память освобождается из-под каждого элемента при его удалении, то есть из-под каждой структурной переменной.

1. Что происходит с элементами стека при его просмотре?

При просмотре стека все элементы из него удаляются, так как для вывода элемента необходимо достать его из стека.

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

Стек эффективнее реализовать с помощью массива, так как он выигрывает во времени обработки стека.  
Хранение с помощью списка может выигрывать, если только стек реализован с помощью статического массива, так как в данном случае размер памяти под список ограничен размером оперативной памяти, а для статического массива — размером стека функции. Значит стек в виде списка может быть большего размера, чем в виде статического массива.