Отчет по лабораторной работе № 4 по дисциплине

“Типы и структуры данных”

Зейналов З. Г. ИУ7-31Б

Вариант 14

Работа № 4

Работа со стеком

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.

Описание условия задачи

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.

Описание ТЗ

Общая концепция системы

Программа выполняет операции над стеком, реализованным двумя способами. Стек вводится пользователем. Сначала вводится количество элементов стека, затем сам стек.

Основная задача - Ввести арифметическое выражение типа:

число|знак| … число|знак| число,

Вычислить значение выражения.

.

Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

* Добавление элемента в стек для двух реализаций
* удаление элемента из стека для двух реализаций
* печать стека в обратном порядке
* вывод времени работы программы
* указание формата и диапазона вводимых данных,
* блокирование ввода данных, неверных по типу,
* указание операции, производимой программой:

1. добавление элемента в стек,
2. удаление элемента из стека,
3. вычисление (обработка данных);

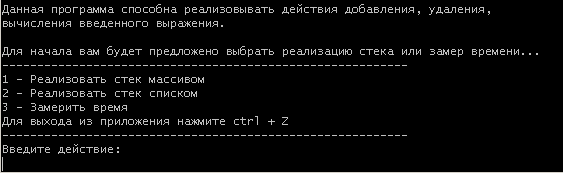
* наличие пояснений при выводе результата.

На вход программа получает действие, связанное с вызовом функций добавления, удаления и вычисления выражения с помощью стека.

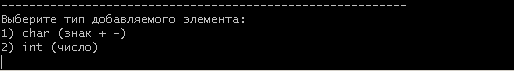
Выход должен быть представлен в числовом формате. Также программа должна вывести данные о времени выполнения алгоритма.

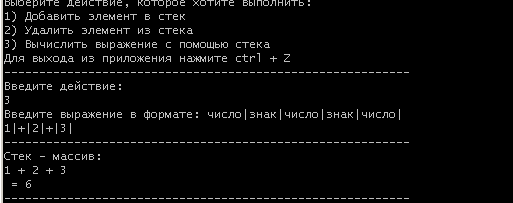
Программа должна выдавать корректный ответ при вводе любых данных. Если произошла ошибка ввода, программа должна сообщить об этом.

Входные данные:



Можно добавлять элементы по типу или ввести выражение для вычисления целиком





Аварийные ситуации

1. Ввод несуществующего пункта меню  
   Программа выведет сообщение “Введенная цифра – неверная, попробуйте еще раз”.
2. Ввод значений, не удовлетворяющих условиям.

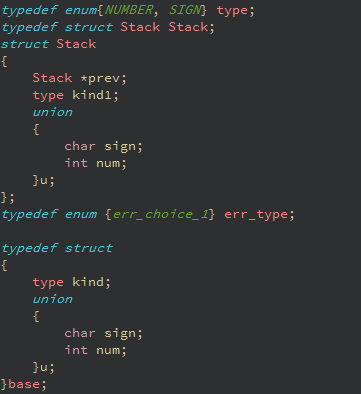
(Ввести вместо знака число или наоборот)

Программа выведет сообщение “ Не могу записать в стек знак, ибо это не знак :(”.

Способ обращения к программе

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

Описание структур данных



Структуры для хранения данных 1- структура – элемент стека  
 2 – структура – элемент массива

Перечисляемые типы type, err\_type для вывода сообщений

void \*array\_mem[ARRAY\_SIZE] – массив для хранения памяти

char number[NUM\_LEN + 1] = ""; чтение числа в строковом формате

base array[ARRAY\_SIZE]; - массив данных

int result = 0; - результат вычисленного выражения

double result\_time = 0;

long int t1, t2, s1\_time = 0, s2\_time = 0; - переменные для подсчета времени выполнения и дальнейшего вычисления

Описание алгоритма

* Пока не выход

1. Добавить элемент   
   Если стек не полон:   
   Добавить элемент в стек   
   Вывести массив занятых адресов
2. Удалить элемент  
   Если стек не пуст:  
   Удалить элемент из стека  
   вывести массив свободных адресов
3. Вывести значение выражения, вычесленного с помощью стека

Вводится выражение  
из стека поэлементно извлекаются элемент

Пока не все элементы

Если число

Записать в инт

Если знак

Записать в строку

Если число

записать в число

Если первое число, второе знак, третье число

Вычислить значение и записать в ответ

Запросить действие еще раз.

Тесты

1.Несуществующий пункт меню  
Вход:

Способ: 4

Вывод: Введенная цифра – неверная, попробуйте еще раз

2. Введен не знак (-, +)

Вход: 22

Вывод: Не могу записать в стек знак, ибо это не знак :(

3. введено не число   
Вход: 123орво

Вывод: 123

4.введено не число

Вход: оывф123

Выход: Не могу записать в стек число, ибо это не число :(

5. удаление элемента из пустого стека

Вход:

2

Вывод: Нет элементов для удаления

6. нормальный вход

вход: 1|+|2|+|3|

Выход: 6

7. Неверный ввод

Вход: 1|+|2|+|

Выход: Неверный ввод, невозможно вычислить выражение

Оценка эффективности

|  |  |
| --- | --- |
| Вычисление выражения из 3 эл. Массив t, такты | Вычисление выражения из 3 эл.  Список t, такты |
| 597 | 5541 |
| Удаление элемента  Массив  t, такты | Удаление элемента  Список  t, такты |
| 43 | 3179 |
| Память на 1 эл стека Массив  в байтах | Память на 1 эл стека Список  В байтах |
| 8 | 16 |

удалось уменьшить объем используемой памяти, использовав объединение в структуре

Выводы по проделанной работе

Вычисление выражения массивом быстрее в 9,28 раз

Удаление элемента массивом быстрее в 73 раза

Память занимаемая элементом массива в 2 раза меньше

Была реализована программа, вычисляющая значение введенного выражения.  
Реализованы функции добавления и удаления элемента для двух способов реализации стека.

Для массива функции удаления и добавления элемента работают быстрее, чем для списка. Для хранения элемента массива используется меньше памяти, чем для хранения структуры. Таким образом, функции работы со стеком, реализованным как массив, эффективнее по памяти и по времени, чем функции для стека-списка.

Ответы на вопросы

1. Что такое стек?

Стек – это последовательный список с переменной длиной, в котором

включение и исключение элементов происходит только с одной стороны – с его

вершины.

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

различной его реализации?

При реализации стека массивом, память выделяется только под элемент массива. (Например, 30 байт для строки из 30 элементов). 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, смещается указатель, память освобождается только в конце работы со стеком.

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?

Эффективней использовать массив при реализации стека, ибо его операции занимают меньше времени