Лабораторная работа №4 по дисциплине

“Типы и структуры данных”

Сорокин Антон ИУ7-32Б

Номер по списку – 24

Вариант 6

***Условие задачи***

Создать программу работы со стеком, выполняющую операции добавления, удаления элементов и вывод 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.

Используя стек, определить, является ли строка палиндромом.

***Техническое задание***

*Исходные данные*

На вход программе подаются последовательно вводимые символы или строка.

*Результат*

Сообщение о том, является ли введённая строка палиндромом или нет.

*Описание задачи*

Программа записывает введённую строку в стек, который реализован как массив или односвязный список (на выбор пользователя); определяет, является ли строка палиндромом; удаляет элемент из стека.

*Аварийное завершение работы программы*

Программа завершается аварийно не может, однако могут аварийно завершиться подпрограммы при:

* пустом стеке;
* переполнении стека как массива;
* ошибке выделения памяти.

***Обращение к программе***

Исполняемый файл app.exe создается путем автоматической сборки проекта с помощью файла makefile. Для выполнения работы следует запустить данный исполняемый файл без указания аргументов.

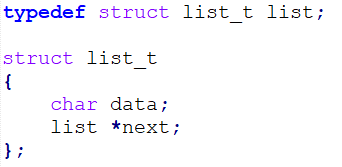
***Алгоритм***

* Ввод символа в стек как массива:
  + Прочитать символ.
  + Проверить, не заполнен ли полностью массив. Если да, то вернуть ошибку.
  + Записать по адресу текущей вершины символ.
  + Передвинуть указатель на вершину на 1 вперёд.
* Ввод символа в стек как списка:
  + Прочитать символ.
  + Проверить, есть ли свободные области памяти.
    - Если да, то взять свободный элемент по адресу из головы списка адресов свободных областей.
    - Если нет, то выделить память под новый элемент списка. Если выделить не удалось, вернуть ошибку.
  + Записать символ в данный элемент списка.
  + Добавить новый элемент в голову списка.
* Удаление символа из стека-массива:
  + Передвинуть указатель на вершину на 1 назад.
* Удаление символа из стека-списка:
  + В списке свободных адресов создать новый элемент. При ошибке выделения памяти вернуть код ошибки.
  + Добавить этот элемент в голову списка.
  + Адрес удаляемого элемента записать в голову списка свободных адресов.
  + Передвинуть голову стека-списка на следующий элемент ***Тестовые данные***

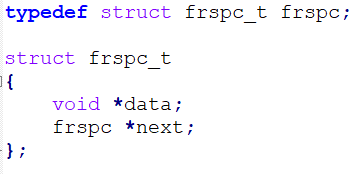
|  |  |  |
| --- | --- | --- |
| Вход | Выход | Класс |
|  | Пустой стек | Пустая строка |
| a | Палиндром | Один символ |
| aaa | Палиндром | Одинаковые символы |
| aabaa | Палиндром | Нечётное кол-во символов, кроме середины, все символы одинаковые |
| babab | Палиндром | Нечётное кол-во символов (палиндром) |
| abba | Палиндром | Чётное кол-во символов (палиндром) |
| acbaa | Не палиндром | Нечётное кол-во символов (палиндром) |
| abca | Не палиндром | Чётное кол-во символов (палиндром) |
| 1001 символ | Переполнение стека | Кол-во символов превышает размер массива |

***Внутренние структуры данных***

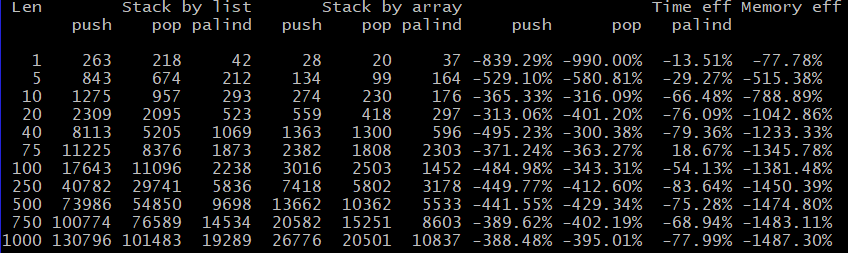
Для хранения стека используется либо односвязный список, либо статический массив. Длина стека хранится в отдельной целочисленной переменной.



Для хранения свободных областей используется односвязный список.



***Оценка эффективности***



Исходя из полученных данных, можно сделать вывод о том, что для реализации стека эффективнее использовать массив. По сравнению с односвязным списком массив эффективнее по памяти, поскольку в списке элемент, помимо данных, должен также хранить указатель на следующий элемент. Такая эффективность зависит от количества элементов. По времени список проигрывает, поскольку тратится дополнительное время на создание или удаления элемента. В среднем работа с массивом как со стеком быстрее 5-6 раз.

***Вывод***

При реализации стека эффективнее использовать статический массив.

***Ответы на вопросы***