Data Structures and Algorithms
LAB #4 - Stacks

Fall 2018

Objectives

After this lab, the student should be able to:

- Use class templates to implement stack (array-based or linked list-based).
- Pass and return Stack (template) to/from functions
- Write code to use stacks in some common applications.

Code Examples

- Open "Lab4.sln"
  1- Run project "***Stack***" and see how stack is implemented as class template using arrays.
  2- Run project "***Passing_Stack***" and see how stack can be passed and returned to/from a function.

Practice Exercises

Exercise 1

The given code example shows how to implement the stack using arrays
You are required to implement the stack the using linked list instead of arrays.
Note: Given class template ***Node*** that you can use to build the linked list-based stack

Exercise 2

In code examples, Project " ***Passing_Stack***"
Update function Reverse to be ***void ReverseMe(…stack to be reversed…)*** so that the function reverses the passed stack itself and does not return anything.

Exercise 3

Write a function “collapse” that takes a stack of integers as a parameter and then collapses it by replacing each successive pair of integers by their sum.

For example:

- suppose a stack stores these values: bottom [**7, 2**, 8, 9, **4, 13**, 7, 1, **9, 10**] top

the stack should become: bottom [**9**, 17, **17**, 8, **19**] top

- and if stack is bottom [1, 2, 3, **4, 5**] top ➔ [1, 5, **9**]

Exercise 4

Write a function **copyStack** that copies the contents of one stack into another in the same order (i.e. not reversed). The function takes two stacks, the source stack and the destination stack.

Exercise 5

A ***palindrome*** is a string that can read backward and forward with the same result.
E.g., the following is a palindrome: **MADAM**
Further, a more general palindrome is one that ignores spacing, punctuation, and capitalization, such as the following is also a palindrome: **Go dog** and **Madam, I'm Adam**
Write a function that takes ***a string stored in a stack of characters*** and checks whether it is a general palindrome or not.

You are allowed to use Stacks only in your functions (char arrays and strings are not allowed)

Exercise 6

One of the applications of a stack is to **backtrack**—that is, to retrace its steps.

- The program read one number at a time until zero is entered or an error occurs (check below)
- Each time a negative number is entered, the program must backtrack and print the five numbers that come before the negative number (starting from the one previous to the negative number) and then discard the negative number.
- If there are fewer than five items in the stack, print an error message and stop the program.
- The program stops normally without an error message when zero is entered.

Test it with the following data:

Input: **80** 1 2 3 4 5 **-1** **20** 30 40 50 **6 7 8 9 10** **-2** 11 12 -3 **22 33** **-5** (press Enter after each number)

Output: 5 4 3 2 1 **10 9 8 7 6** 12 11 50 40 30 **22 33 20 80** Error: less than 5 items in the stack