Implement two stacks in an array

Create a data structure twoStacks that represents two stacks. Implementation of twoStacks should use only one array, i.e., both stacks should use the same array for storing elements. Following functions must be supported by twoStacks.
push1(int x) –> pushes x to first stack
push2(int x) –> pushes x to second stack
pop1() –> pops an element from first stack and return the popped element
pop2() –> pops an element from second stack and return the popped element
Implementation of twoStack should be space efficient.

##

Design a stack with operations on middle element
How to implement a stack which will support following operations in O(1) time complexity?
1) push() which adds an element to the top of stack.
2) pop() which removes an element from top of stack.
3) findMiddle() which will return middle element of the stack.
4) deleteMiddle() which will delete the middle element.
Push and pop are standard stack operations.

##

Check for Balanced Brackets in an expression (well-formedness) using Stack

Given an expression string exp, write a program to examine whether the pairs and the orders of “{“, “}”, “(“, “)”, “[“, “]” are correct in exp.

Example:

Input: exp = “[()]{}{[()()]()}”
Output: Balanced

Input: exp = “[(])”
Output: Not Balanced