

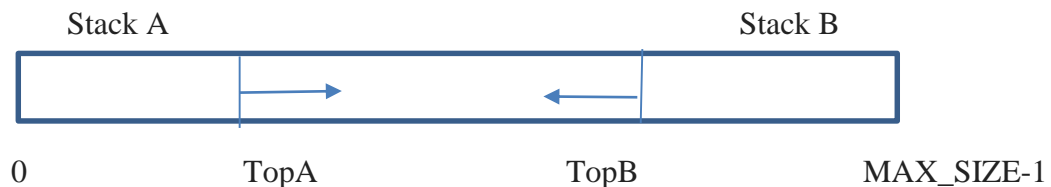
Lab Assignment #2

1. Create an ADT Stack_Y15. It contains the name of students of your class. The ADT stack is specified by the following operations:

- push (x): inserts a student name x.
- x = pop (): delete the name of student from the top of stack and return in x.
- i = isempty (): check whether the stack is empty or not? Function return 1 if empty otherwise 0.

Implement this Stack_Y15 using the data structure "Array" and "Linked List, separately. Write a menu driven program for users to operate on this stack. We have discussed the implementation of stack containing integers. Now try for implementing the above one which contains the character arrays.

2. Suppose we need to maintain two stacks of the same type of items in a program. If the two stacks are stored in separate arrays, then one stack might overflow while there are considerable unused space in other. To avoid this situation it is better to maintain two stacks in the same array as shown below:



Let the size of the array is MAX_SIZE. In this structure, one stack say A, grows from one end of the array i.e., index 0 and the other stack say B, starts at the other end i.e., index MAX_SIZE-1 and grows in the opposite direction that is towards the direction of A.

Write a program in C to implement the above mentioned double stack structure for the following operations:

PUSHA: push element in stack A.

PUSHB: push element in stack B

POPA: Pop element from stack A.

POPB: Pop element from stack B.

ISEMPTYA: Check the stack A is empty or not?

ISEMPTYB: Check the stack B is empty or not?

Assume that the stacks are containing integer items only.

3. Implement stack applications:

- Balanced parenthesis. Check for all three kinds of brackets: { }, (), [].
- Evaluation of postfix expression containing the operator: \$, +, -, *, /, =, >, <, !.
- Conversion from infix to postfix.
- Conversion from infix to prefix.
- Reverse a string.