

DSA LAB Programming Practice

STACK

1. Implement a stack using dynamic array & linked list and define following methods -
 1. initialize
 2. Push
 3. Pop
 4. IsFull
 5. IsEmpty
2. Write a program to display the elements of a single linked list in reverse way using STACK with order of n complexity. (You do not have to use recursive function)
3. Write a program to check whether a string is bracket (i.e. [], {} and ()) balanced or not using stack. (For example, the string “((()())())” contains properly nested pairs of parentheses, but the string “()()” does not, and the string “())” does not contain properly matching parentheses.)
4. Write a function to convert Infix expression to Postfix expression.
5. Write a function to evaluate postfix expression and prefix expression.
6. Write program copy one STACK to another STACK without changing the order of its elements.
7. Implement undo and redo operation on array’s insertion and deletion operation using STACK.
8. Write a program to find the duplicate parentheses of an expression and remove the duplicate expression.
Ex: ((a+b))*c
(a+b) * c
9. Write a program to implement a stack that will return minimum element in constant time.
10. Write a program to find the longest increasing sub-sequence.
Input: [10,9,2,5,3,7,101,18]
Output: 4
Explanation: The longest increasing sub-sequence is [2,3,7,101], therefore the length is 4.