Assignment-55 : A Job Ready Bootcamp in C++, DSA and IOT

Stack

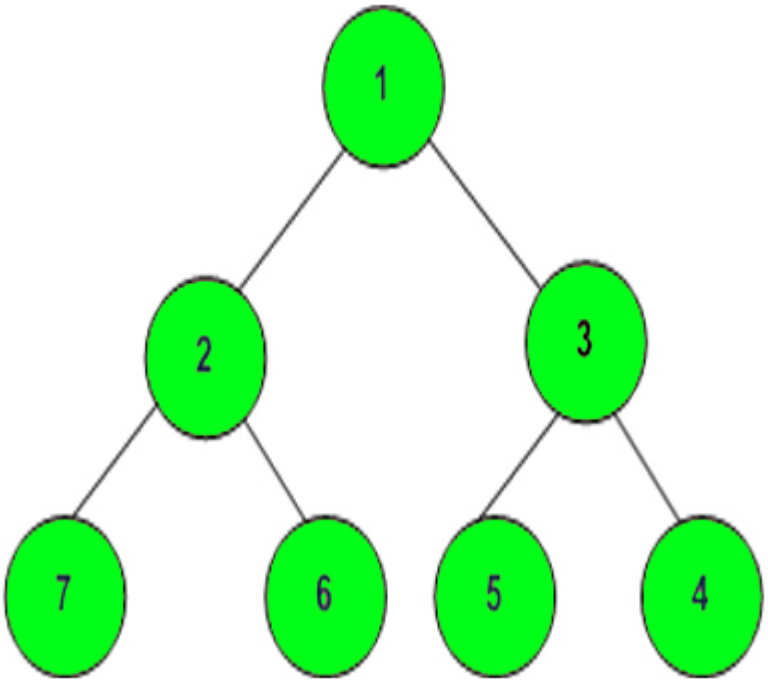
1. Given a stack with push(), pop(), empty() operations, delete the middle of the stack without using any additional data structure.
   Middle: ceil((size_of_stack+1)/2) (1-based index)
   Example 1:

   Input:
   Stack = {1, 2, 3, 4, 5}
   Output:
   ModifiedStack = {1, 2, 4, 5}
   Explanation:
   As the number of elements is 5, hence the middle element will be the 3rd element which is deleted

   Example 2:
   Input:
   Stack = {1 2 3 4}
   Output:
   ModifiedStack = {1 3 4}
   Explanation:
   As the number of elements is 4, hence the middle element will be the 2nd element which is deleted

2. Given an expression string x. Examine whether the pairs and the orders of “{“,”}”,”(“,”)”,”[“,”]” are correct in exp.
   For example, the function should return 'true' for exp = “[()]{}{[()()]()}” and 'false' for exp = “[(])”.

3. Complete the function to find spiral order traversal of a tree. For the tree below, function should return 1, 2, 3, 4, 5, 6, 7.



Example 1:
Input:
```
      1
   /     \
  3       2
```
Output:1 3 2

Example 2:

Input:
```
              10
           /     \
         20      30
        /  \
      40   60
```
Output: 10 20 30 60 40

4. Given a stack, the task is to sort it such that the top of the stack has the greatest element.
   Example 1:
   Input:
   Stack: 3 2 1
   Output: 3 2 1
   Example 2:

Input:
Stack: 11 2 32 3 41
Output: 41 32 11 3 2

5. Reverse the string using stack.

6. Given a string S, the task is to find the bracket numbers.
   Example 1:
   Input: S = "(aa(bdc))p(dee)â€‹"
   Output: 1 2 2 1 3 3
   Explanation: The highlighted brackets in the given string (aa(bdc))p(dee) has been assigned the numbers as: 1 2 2 1 3 3.
   Example 2:
   Input: S = "(((()("
   Output: 1 2 3 4 4 5
   Explanation: The highlighted brackets in
   the given string (((((has been assigned
   the numbers as: 1 2 3 4 4 5

7. Design a data-structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. Your task is to complete all the functions, using stack data-Structure.
   Example 1:
   Input:
   Stack: 18 19 29 15 16
   Output: 15
   Explanation:
   The minimum element of the stack is 15.

8. You are given the string S . Compress the string when lower and upper cases are the same. In compressed string characters should be in lowercase.
   Example 1:
   Input: S = "aaABBb"
   Output: "3a3b"
   Explanation: As 'a' appears 3 times consecutively and 'b' also 3 times, and 'b' and 'B' are considered the same.
   Example 2:
   Input: S = "aaacca"
   Output: "3a2c1a"
   Explanation: As 'a' appears 3 times consecutively and 'c' also 2 times, and then 'a' 1 time.

9. Given two strings s and t, return true if they are equal when both are typed into empty text editors. '#' means a backspace character. Note that after backspacing an empty text, the text will continue empty.

   Example 1:

Input: s = "ab#c", t = "ad#c"
Output: true
Explanation: Both s and t become "ac".
Example 2:
Input: s = "ab##", t = "c#d#"
Output: true
Explanation: Both s and t become "".
Example 3:
Input: s = "a#c", t = "b"
Output: false
Explanation: s becomes "c" while t becomes "b".

10. Your task is to implement 2 stacks in one array efficiently.

Example 1:
Input:
enqueue(2)
enqueue(3)
dequeue()
enqueue(4)
dequeue()
Output: 2 3
Explanation:
enqueue(2) the queue will be {2}
enqueue(3) the queue will be {3 2}
dequeue() the poped element will be 2
the stack will be {3}
enqueue(4) the stack will be {4 3}
dequeue() the poped element will be 3.
Example 2:
Input:
enqueue(2)
dequeue()
dequeue()
Output: 2 -1