

Data Structure and Algorithms

Duration: 2 Hours

Note : All coding must be written only in Python

Answer all six questions .

Each Question carries 5 marks.

Max Marks : 30

Q1:

A sequence of brackets is balanced if the following conditions are met:

It contains no unmatched brackets. The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets. Given n strings of brackets, determine whether each sequence of brackets is balanced. If a string is balanced, return YES. Otherwise, return NO.

Function Description

Complete the function `isBalanced` .

`isBalanced` has the following parameter(s):

string `s`: a string of brackets

Returns

string: either YES or NO

Q2:

A left rotation operation on an array of size n shifts each of the array's elements 1 unit to the left. Given an integer, d , rotate the array that many steps left and return the result.

Example

$d=2$

`arr=[1,2,3,4,5]`

After 2 rotations,

`arr'=[3,4,5,1,2]` .

Function Description

Complete the `rotateLeft` function in the editor below.

`rotateLeft` has the following parameters:

`int d`: the amount to rotate by

`int arr[n]`: the array to rotate

Q3:

Given a pointer to the head node of a linked list, print each node's data element, one per line. If the head pointer is null (indicating the list is empty), there is nothing to print.

Function Description

Complete the `printLinkedList` function in the editor below.

`printLinkedList` has the following parameter(s):

`SinglyLinkedListNode head`: a reference to the head of the list

Print For each node, print its data value on a new line

Returns

`int[n]`: the rotated array

Q4:

Complete the `PreOrder` function, which has 1 parameter: a pointer to the root of a binary tree. It must print the values in the tree's preorder traversal as a single line of space-separated values.

Q5:

Complete the `getHeight` or `height` function . It must return the height of a binary tree as an integer. `getHeight` or `height` has the following parameter(s):

- `root`: a reference to the root of a binary tree.

Q6:

Given the pointer to the head node of a linked list, change the next pointers of the nodes so that their order is reversed. The head pointer given may be null meaning that the initial list is empty.

Example

head references the list $1 \rightarrow 2 \rightarrow 3 \rightarrow \text{NULL}$

Manipulate the next pointers of each node in place and return head, now referencing the head of the list $3 \rightarrow 2 \rightarrow 1 \rightarrow \text{NULL}$.

Function Description

Complete the reverse function in the editor below.

reverse has the following parameter:

- SinglyLinkedListNode pointer head: a reference to the head of a list

Returns

- SinglyLinkedListNode pointer: a reference to the head of the reversed list

Note: Coding must be uploaded In the GitHub under your name before 5pm. Students who submit after 5pm will not be considered.