

CDAC Mumbai
PG-DAC Mar22 Batch (Juhu & Kharghar)
Module 3: Algorithms & Data Structures

Time : 2hrs

Marks : 30

Q1: Sorting

Assume you are given the array $a1=[1,2,4,5,3]$ indexed $0 \dots 4$. Store the value of $a1[4]$. Now test lower index values successively from 3 to 0 until you reach a value that is lower than $a1[4]$, at $a1[1]$ in this case. Each time your test fails, copy the value at the lower index to the current index and print your array. When the next lower indexed value is smaller than $a1[4]$, insert the stored value at the current index and print the entire array.

Example

$n=5$

$a1 = [1,2,4,5,3]$

Start at the rightmost index. Store the value of $a1[4]=3$. Compare this to each element to the left until a smaller value is reached. Here are the results as described:

1 2 4 5 5

1 2 4 4 5

1 2 3 4 5

Function Description

Complete the *insertionSort* function in the editor below.

insertionSort has the following parameter(s):

n : an integer, the size of $a1$

$a1$: an array of integers to sort

Returns

Note: Print the interim and final arrays, each on a new line. No return value is expected.

Input Format

The first line contains the integer n , the size of the array $a1$.

The next line contains n space-separated integers $a1[0] \dots a1[n-1]$.

Constraints

$1 \leq n \leq 1000$

$-1000 \leq a1[i] \leq 10000$

Output Format

Print the array as a row of space-separated integers each time there is a shift or insertion.

Test case 1:

5

2 4 6 8 3

Sample Output

2 4 6 8 8

2 4 6 6 8

2 4 4 6 8

2 3 4 6 8 [10 marks]

Q2:Reverse a Linked List

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->2->3->NULL

Manipulate the next pointers of each node in place and return head, now referencing the head of the list.

3->2->1->NULL.

Function Description

Complete the *reverse* function in the editor below.

reverse has the following parameter:

- *Node pointer head*: a reference to the head of a list

Returns

LinkedListNode pointer: a reference to the head of the reversed list

Input Format

The first line contains an integer *t*, the number of test cases.

Each test case has the following format:

The first line contains an integer *n*, the number of elements in the linked list.

Each of the next *n* lines contains an integer, the data values of the elements in the linked list.

Test case 1:

1

5

1

2

3

4

5

Output :

5 4 3 2 1

Test case 2:

2

4

3

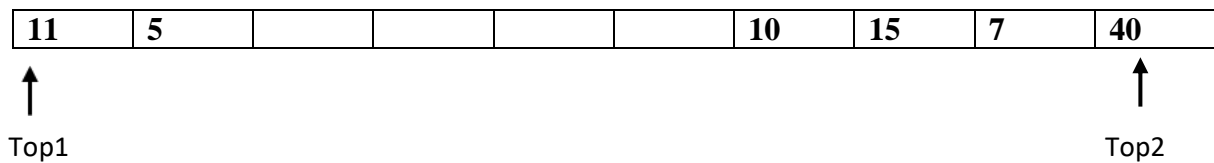
4

2

5

Output:

5 2 4 3

[10 marks]**Q3:Java program to implement two stacks in a single array.****Note: Please check is required to be update in list and then define test case.**

Input values :

```
push1(5);  
push2(10);  
push2(15);  
push1(11);  
push2(7);  
push2(40);
```

Output: **(Required to be printed like this only)**

Popped element from stack1 is 11

Popped element from stack2 is 40

[10 marks]