### Sort a stack

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

**Expected Time Complexity**: O(N\*N)  
**Expected Auxilliary Space**: O(N) recursive.

**Constraints:**  
1<=N<=100

**Company Tags**

[**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Goldman Sachs**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Goldman%20Sachs)[**Intuit**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Intuit) [**Yahoo**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Yahoo) [**IBM**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=IBM) [**Kuliza**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Kuliza)

**Topic Tags**

[**Recursion**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Recursion) [**Stack**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Stack) [**STL**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=STL) [**Data Structures**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Data%20Structures) [**Algorithms**](https://practice.geeksforgeeks.org/explore/?category%5b%5d=Algorithms)

//{ Driver Code Starts

import java.util.Scanner;

import java.util.Stack;

class SortedStack{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int t=sc.nextInt();

while(t-->0){

Stack<Integer> s=new Stack<>();

int n=sc.nextInt();

while(n-->0)

s.push(sc.nextInt());

CodingMaxima g=new CodingMaxima ();

Stack<Integer> a=g.sort(s);

while(!a.empty()){

System.out.print(a.peek()+" ");

a.pop();

}

System.out.println();

}

}

}

class CodingMaxima{

public Stack<Integer> sort(Stack<Integer> s)

{

int k=0;

int[] ar=new int[s.size()] ;

while(!s.isEmpty())

ar[k++]=(s.pop());

java.util.Arrays .sort(ar);

for( int num :ar)

s.push(num);

return s;

}

}