### k largest elements

**Company Tags**

[**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Samsung**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Samsung) walmart **[Google](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Google" \t "_blank)**

Given an array **Arr** of **N** positive integers and an integer **K**, find **K** **largest elements**from the array.  The output elements should be printed in decreasing order.

**Example 1:**

**Input:**

N = 5, K = 2

Arr[] = {12, 5, 787, 1, 23}

**Output:** 787 23

**Explanation:** 1st largest element in the

array is 787 and second largest is 23.

**Example 2:**

**Input:**

N = 7, K = 3

Arr[] = {1, 23, 12, 9, 30, 2, 50}

**Output:** 50 30 23

**Explanation:** 3 Largest element in the

array are 50, 30 and 23.

### Java solution

//{ Driver Code Starts

import java.util.\*;

import java.io.\*;

public class Main {

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int tc = Integer.parseInt(br.readLine().trim());

while (tc-- > 0) {

String[] inputLine;

inputLine = br.readLine().trim().split(" ");

int n = Integer.parseInt(inputLine[0]);

int k = Integer.parseInt(inputLine[1]);

int[] arr = new int[n];

inputLine = br.readLine().trim().split(" ");

for (int i = 0; i < n; i++) {

arr[i] = Integer.parseInt(inputLine[i]);

}

int[] ans = new Solution().kLargest(arr, n, k);

for (int x : ans) {

System.out.print(x + " ");

}

System.out.println();

}

}

}

// } Driver Code Ends

class Solution {

int[] kLargest(int[] arr, int n, int k) {

int[] ans=new int[k];

Arrays.sort(arr);

for(int i=0;i<k;i++)

ans[i]=arr[n-i-1];

return ans;

}

}