**Second Largest**

[array](http://www.practice.geeksforgeeks.org/tag-page.php?tag=array&isCmp=0)[sorting](http://www.practice.geeksforgeeks.org/tag-page.php?tag=sorting&isCmp=0)[Rockstand](http://www.practice.geeksforgeeks.org/tag-page.php?tag=Rockstand&isCmp=1)

Given an array, return second last element from an array after the sorting of the array.

**Input:**

The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is N,N is the size of array.  
The second line of each test case contains N input C[i].  
  
**Output:**

Print the second largest element.  
  
**Constraints:**

1 ≤ T ≤ 30  
1 ≤ N ≤ 100  
1 ≤ C[i] ≤ 1000  
  
**Example:**

Input  
2  
5  
89 24 75 11 23

6

56 42 21 23 56 20

Output  
75

56

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=538>

#include <iostream>

#include <stdio.h>

#include <set>

#include <map>

#include <vector>

using namespace std;

int main() {

    // TODO code application logic here

    int t;

    scanf("%d", &t);

    while(t-- > 0) {

       int n;

       scanf("%d", &n);

       std::vector<int> v;

       for(int i =0 ; i < n; i++) {

          int elem;

          scanf("%d", &elem);

          v.push\_back(elem);

       }

       std::sort(v.begin(), v.end());

       int index = v.size() - 1;

       while ( index > 0 && v[index] == v[index-1]){

            index--;

       }

       printf("%d**\n**", v[index - 1]);

    }

}