**Sort in specific order**

Show Topic Tags   

[Microsoft](http://practice.geeksforgeeks.org/company/Microsoft/)

Given an array of integers (both odd and even), the task is to sort them in such a way that the first part of the array contains odd numbers sorted in descending order, rest portion contains even numbers sorted in ascending order.

Examples:

Input : A[] = {1, 2, 3, 5, 4, 7, 10}

Output : A[] = {7, 5, 3, 1, 2, 4, 10}

Input : A[] = {0, 4, 5, 3, 7, 2, 1}

Output : A[] = {7, 5, 3, 1, 0, 2, 4}

**Input:**  
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains an integer N denoting the size of the array. The next line contains N space separated values of the array.  
  
**Output:**  
For each test case in a new line print the space separated values of the  new transformed array.  
  
**Constraints:**  
1<=T<=100  
1<=N<=100  
1<=A[]<=100  
  
**Example:  
Input:**  
2  
7  
1 2 3 5 4 7 10  
7  
0 4 5 3 7 2 1  
**Output:**  
7 5 3 1 2 4 10  
7 5 3 1 0 2 4

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

New to competitive programming? Please see [How to Begin?](http://www.geeksforgeeks.org/how-to-begin-with-competitive-programming/)and [How to Pick a Category?](http://practice.geeksforgeeks.org/pickACategory.php)

<http://practice.geeksforgeeks.org/problems/sort-in-specific-order/0>

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class GFG {

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

// TODO code application logic here

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

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

String[] input = br.readLine().trim().split(" ");

int[] arr = new int[n];

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

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

}

List<Integer> pares = new ArrayList<Integer>();

List<Integer> impares = new ArrayList<Integer>();

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

if(arr[i]%2==0) {

pares.add(arr[i]);

}else{

impares.add(arr[i]);

}

}

Collections.sort(impares);

//Collections.reverse(impares);

Collections.sort(pares);

for(int i =impares.size()-1 ; i>=0; i--){

System.out.print(impares.get(i) + " ");

}

for(int i =0; i<pares.size(); i++) {

System.out.print(pares.get(i) + " ");

}

System.out.println();

}

}

}