**Rearranging array**

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

Given a list of integers, rearrange the list such that it consists of alternating minimum maximum elements **using only list operations**. The first element of the list should be minimum and second element should be maximum of all elements present in the list. Similarly, third element will be next minimum element and fourth element is next maximum element and so on. Use of extra space is not permitted

**Input:**  
The first line of input contains a single integer T denoting the number of test cases. ThenT test cases follow. Each test case consist of two lines. The first line of each test case consists of an integer N, where N is the size of array.The second line of each test case contains N space separated integers denoting array elements.

**Output:**  
Corresponding to each test case, in a new line, print the modified list.

**Constraints:**  
1 ≤ T ≤ 100  
1 ≤ N ≤ 200  
1 ≤ A[i] ≤ 500  
  
**Example:**

**Input**  
1  
5  
4 5 1 2 3

**Output**  
1 5 2 4 3

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

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=1325>

#include <stdio.h>

#include <math.h>

#include <iostream>

#include <cmath>

#include <vector>

#define ll long long int

using namespace std;

int main()

{

int t;

scanf("%d",&t);

while(t--) {

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());

std::vector<int> ans;

int i =0, j=n-1;

while(i<j) {

ans.push\_back(v[i]);

ans.push\_back(v[j]);

i++;

j--;

}

if(i == j) {

ans.push\_back(v[i]);

}

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

printf("%d ", ans[i]);

}

printf("\n");

}

return 0;

}