**Even and odd elements at even and odd positions**

Show Topic Tags   

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

Given an array. The task is to arrange the array such that odd elements occupy the odd positions and even elements occupy the even positions. The order of elements must remain same. Consider zero-based indexing. After printing according to conditions, if remaining, print the remaining elements as it is.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case consists of two lines. First line of each test case contains an Integer N denoting size of array and the second line contains N space separated elements.

**Output:**  
For each test case, in a new line print the arranged array.

**Constraints:**  
1<=T<=100  
1<=N<=105  
1<=A[i]<=105

**Example:  
Input:**  
2  
6  
1 2 3 4 5 6  
4  
3 2 4 1  
**Output:**  
2 1 4 3 6 5  
2 3 4 1

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

Contributor: Arun Tyagi

<http://practice.geeksforgeeks.org/problems/even-and-odd-elements-at-even-and-odd-positions/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication250;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.math.BigInteger;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.HashMap;

import java.util.HashSet;

import java.util.LinkedHashSet;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication250 {

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().trim());

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

ArrayList<Integer> pares = new ArrayList();

ArrayList<Integer> impares = new ArrayList();

int i = 0;

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

{

int elem = Integer.parseInt(input[i]);

if (elem % 2 == 0)

{

pares.add(elem);

}

else

{

impares.add(elem);

}

}

//int[] arr = new int[n];

ArrayList<Integer> lista = new ArrayList<Integer>();

int a = 0;

int b = 0;

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

{

if (i % 2 == 0)

{

if (a < pares.size())

{

lista.add(pares.get(a));

a++;

}

}

else if (i % 2 != 0)

{

if (b < impares.size())

{

lista.add(impares.get(b));

b++;

}

}

}

while (a < pares.size())

{

lista.add(pares.get(a));

a++;

if (i + 1 < n)

{

i++;

}

}

while (b < impares.size())

{

lista.add(impares.get(b));

b++;

if (i + 1 < n)

{

i++;

}

}

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

{

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

}

System.out.println();

}

}

}