**Inverse List**

Attempted by: **1123**

/

Accuracy: **33%**

/

Maximum Score: **20**

/

6 Votes

Tag(s):

Ad-Hoc, Easy, Implementation

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

There are many ways to order a list of integers from 11 to nn. For example, if n=3n=3, the list could be : [312][312].

But there is a special way to create another list from the given list of integers. In this list, position of integer ii is the i−thi−th number in the given list. So following this rule, the given list will be written as: [231][231]. This list is called inverse list. Now there exists some list whose inverse list is identical. For example, inverse list of [123][123] is same as given list. Given a list of integers you have to determine whether the list is inverse or not.

The input contains several test cases. The first line is the number of test cases t (1<=t<=100)(1<=t<=100) . The first line of each test case contains an integer n(1<=n<=100000)n(1<=n<=100000). Then a list of the integers 11 to nn follows in the next line.

**SAMPLE INPUT**

2

3

3 1 2

3

1 2 3

**SAMPLE OUTPUT**

not inverse

inverse

**Time Limit:**2.0 sec(s) for all input files combined.

**Memory Limit:**256 MB

**Source Limit:**1024 KB

**Marking Scheme:**Marks are awarded when all the testcases pass.

**Allowed Languages:**C, C++, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, JavaScript(Rhino), JavaScript(Node.js), Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, R(RScript), Racket, Ruby, Rust, Scala, Scala 2.11.8, Swift, Visual Basic

<https://www.hackerearth.com/practice/basic-programming/implementation/basics-of-implementation/practice-problems/algorithm/inverse-list/>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication2

{

class Program

{

static void Main(string[] args)

{

//int[] a = Array.ConvertAll("76 33 65 40 85 52 91 25 27 87 57 45 86 9 69 10 13 46 68 23 84 37 14 19 28 38 18 50 48 43 80 6 81 70 75 61 88 32 8 20 34 59 54 47 1 5 56 17 60 89 82 31 55 11 71 77 42 64 7 63 78 2 53 29 74 66 79 3 16 67 39 26 73 51 41 15 90 22 35 58 62 4 83 21 12 72 30 36 49 44 24".Split(' '), e => int.Parse(e));

//int[] a = Array.ConvertAll("2 1 4 3".Split(' '), e => int.Parse(e));

//Console.WriteLine(Array.IndexOf(a, 2));

//int[] a = Array.ConvertAll("1 3 2".Split(' '), e => int.Parse(e));

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

int n = int.Parse(Console.ReadLine());

int[] a = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

Dictionary<int, int> indices = new Dictionary<int, int>();

for (int i = 0; i < a.Length; i++)

{

indices[a[i]] = i + 1;

}

string ans = "inverse";

foreach (KeyValuePair<int, int> kvp in indices)

{

// Console.WriteLine(kvp.Key + " " + kvp.Value);

if (indices[kvp.Value] != kvp.Key)

{

ans = ("not inverse");

break;

}

}

Console.WriteLine(ans);

}

Console.ReadLine();

}

}

}