COMPETITIVE CODING
with
RAMAN CLASSES

RAMAN CLASSES

FULFILL YOUR DREAMS OF SUDYING IN IIT's

... An interactive series for those willing to learn and code

Problem :Beautiful And Ugly

Topic	: **Arrays**
Difficulty	: **EASY**
Programming Language	: **C++**
Time to Spend	: **15 min**

Problem:
You are given an array, a , of n non-zero positive integers. The array is said to be beautiful if all the following constraints are satisfied:

1. The array consists of unique elements.
2. The array elements are not sorted in ascending order.
3. All the array elements should have a value
    between 1 to n inclusive, i.e.,$1 <= a_i <= n$, where, $0 <= i < n$.

If the array is beautiful, print Beautiful; otherwise print Ugly.

Problem Statement

Input :
5
1 2 3 4 5

Output :
Ugly

Input Description :
The first line of the input is an integer q , the total number of queries. Each query consists of two lines.

The first line of each query contains an integer n denoting the total number of elements in the array and the second line of each query contains n space separated integers describing the array a .

Output Description :

For each query, print Beautiful if the array is beautiful; otherwise print Ugly on a new line.

In order to solve this problem, go through the following concepts.

1. Loop Statements
2. Conditional Statements

Problem Description

We are given an array a of n size and we need to find whether the array is ugly or beautiful according to the conditions which are
1. Each array element should be unique
2. Array should not be in ascending order
3. The element should be in range of 1 to n.

For example: a=[5,4,3,2,1]
Here, every element is unique and array is not in ascending order and all the elements are between 1 to 5 , that's why it is beautiful.

Output: Beautiful

Let Us Think

In order to solve this problem, let us think and analyse how to get started with this problem.

Try to draw different examples on paper. On careful observation, you can see we have to make use of suitable conditional statements in the one pass of array to find out whether it is beautiful or not.

Let Us Think

Now you know the logic. Lets proceed with the code.

Now you know the logic. Lets proceed with the code.

Things we need to do for this problem:
1. Write code for the condition of presence of only unique element.

2. Write code for the condition that array is not sorted in ascending order.

3. Write code to check if the element is in between 1 to n or not.

1. Write code for the condition of presence of only unique element.

```
// Checking if the array has unique element or not.

if (s.find(a[i]) == s.end()) {
    s.insert(a[i]);
}
else {
    flag1 = 1;
}
```

Here, the flag1 is initially set to 0, when the condition of not having unique element fails, it gets updated to 1.

2. Write code to check if the element is in between 1 to n or not.

```
// Checking if the array element is in between 1 to n or not.

if (a[i] < 1 || a[i] > n + 1) {
    flag1 = 1;
}
```

Here, the flag1 is initially set to 0, when the condition of having element in between 1 to n is not satisfied, it gets updated to 1.

3. Write code for the condition that array is not sorted in ascending order.

```
// Checking if the array element are sorted in ascending or not.
if (i + 1 < n) {
    if (a[i] > a[i + 1]) {
        flag2 = 0;
    }
}
```

Here, the flag2 is initially set to 1 assuming that array is sorted in ascending order, which gets updated to 0, when the assumption fails.

Thank You !

... brought to you by

www.ramanclasses.in