Gaming Array

locked

Andy wants to play a game with his little brother, Bob. The game starts with an array of distinct integers and the rules are as follows:

- Bob always plays first and the two players move in alternating turns.
- In a single move, a player chooses the maximum element currently present in the array and removes it as well as all the other elements to its right. For example, if the starting array $arr = [2, 3, 5, 4, 1]$, then it becomes $arr' = [2, 3]$ after the first move because we remove the maximum element (i.e., 5) and all elements to its right (i.e., 4 and 1).
- The modifications made to the array during each turn are permanent, so the next player continues the game with the remaining array. The first player who is unable to make a move loses the game.

Andy and Bob play g games. Given the initial array for each game, find and print the name of the winner on a new line. If Andy wins, print `ANDY`; if Bob wins, print `BOB`.

To continue the example above, in the next move Andy will remove 3. Bob will then remove 2 and win because there are no more integers to remove.

Function Description

Complete the *gamingArray* function in the editor below. It should return a string that represents the winner, either `ANDY` or `BOB`.

gamingArray has the following parameter(s):

- *arr*: an array of integers

Input Format

The first line contains a single integer g, the number of games.

Each of the next g pairs of lines is as follows:

- The first line contains a single integer, n, the number of elements in arr.
- The second line contains n distinct space-separated integers $arr[i]$ where $0 \leq i < n$.

Constraints

- Array arr contains n distinct integers.

For 35% of the maximum score:

- $1 \leq g \leq 10$
- $1 \leq n \leq 1000$
- $1 \leq arr[i] \leq 10^5$
- The sum of n over all games does not exceed 1000.

For 100% of the maximum score:

- $1 \leq g \leq 100$

- $1 \leq n \leq 10^5$
- $1 \leq a_i \leq 10^9$
- The sum of n over all games does not exceed 10^5.

Output Format

For each game, print the name of the winner on a new line (i.e., either `BOB` or `ANDY`).

Sample Input 0

```
2
5
5 2 6 3 4
2
3 1
```

Sample Output 0

```
ANDY
BOB
```

Explanation 0

Andy and Bob play the following two games:

1. Initially, the array looks like this:

In the first move, Bob removes 6 and all the elements to its right, resulting in $A = [5, 2]$:

In the second move, Andy removes 5 and all the elements to its right, resulting in $A = []$:

At this point, the array is empty and Bob cannot make any more moves. This means Andy wins, so we print `ANDY` on a new line.

2. In the first move, Bob removes 3 and all the elements to its right, resulting in $A = []$. As there are no elements left in the array for Andy to make a move, Bob wins and we print `BOB` on a new line.

Sample Input 1

```
2
5
1 3 5 7 9
5
7 4 6 5 9
```

Sample Output 1

```
BOB
ANDY
```

Explanation 1

In the first test, they alternate choosing the rightmost element until the end. Bob, Andy, Bob, Andy, Bob.

In the second case, Bob takes 9, Andy takes $[7, 4, 6, 5]$.

Submissions: 0
Max Score: 0
Difficulty: Medium

Rate This Challenge:

More

Current Buffer (saved locally, editable)

C++

```
#include<bits/stdc++.h>
#include<stdio.h>

int main()
{

    int t;
    std::cin>>t;
    while(t--)
    {
        int n;int m=0;
        std::cin>>n;
        long int max=0;
        for(int i=0;i<n;i++)
        {
            long int x;
            std::cin>>x;
            if(x>max)
            {
                max=x;
                ++m;
            }
        }
```

```
24
25
26
27
28              if(m%2==0)
29                  std::cout<<"ANDY"<<"\n";
30              else
31                  std::cout<<"BOB"<<"\n";
32      }
33
34
35
36      return 0;
37  }
```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔ Testcase 1 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
2
5
5 2 6 3 4
```

```
2
3 1
```

Your Output (stdout)

```
ANDY
BOB
```

Expected Output

```
ANDY
BOB
```