https://course.acciojob.com/idle?question=12a72266-1582-45fe-9d30-db3a1a0b710e

- MEDIUM
- Max Score: 40 Points
-

Floor in a Sorted Array

Given a sorted array arr[] of size N without duplicates, and given a value x. Find the index of floor of x in given array. Floor of x is defined as the largest element K in arr[] such that K is smaller than or equal to x.

Try to use binary search to solve this problem.

Input Format

- **First line of input contains number of integers in array, N and element whose floor is to be searched.**
- **Last line of input contains array elements.**

Output Format

Output the index of floor of x if exists, else print -1. Use 0-indexing.

Example

Input

```
7 0
1 2 8 10 11 12 19
```

Output

```
-1
```

Explanation

No element less than or equal to 0 is found. So output is "-1".

Example 2

Input

```
7 5
1 2 8 10 11 12 19
```

Output

```
1
```

Explanation

Number less than or equal to 5 is 2, whose index is 1(0-based indexing).

Example 3

Input

```
7 10
1 2 8 10 11 12 19
```

Output

```
3
```

Explanation

Number less than or equal to 10 is 10 and its index is 3.

Constraints

- **1 ≤ N ≤ 10^5**
- **1 ≤ arr[i] < 10^9**
- **0 ≤ X ≤ arr[n-1]**

Topic Tags

- **Binary Search**

My code

```
// n java
import java.util.*;
import java.io.*;

public class Main {
    public static void main(String args[]) {
        //your code here
            Scanner s=new Scanner(System.in);
            int n=s.nextInt();
            int k=s.nextInt();
            int arr[]=new int[n];
            for(int i=0;i<n;i++)
              arr[i]=s.nextInt();
            int lp=0,rp=n;
           while(lp<=rp)
                    {
                            int mid=(lp+rp)/2;
                            if(arr[mid]==k)
                            {
                                    System.out.print(mid);
                                    return;
                            }
                            else if(arr[mid]>k)
                                    rp=mid-1;
                            else lp=mid+1;
```

```
            }
        System.out.print(lp-1);
        //System.out.print("Hl");
    }
}
```