https://course.acciojob.com/idle?question=127fb51b-1c5d-4e3a-a2db-7e6ae72fe535

- EASY
- Max Score: 30 Points

Floor And Ceil

Given an sorted integer array `arr` of size `n` which contains unique elements, find the floor and ceil of a given `key`.

If the `key` is present in array floor and ceil of that key is itself that number and if not then simply find its floor and ceil of the `key` in the array.

Note

If the ceiling/floor of the `key` is not present in the array return `-1`.

Input Format

First line contains two spaced integers the array size `n` and `key`

Next line contains `n` spaced integers.

Output Format

Return an array containing floor and ceil of the given `key` and make sure you have floor first and then ceil in the result array.

Example 1

Input

```
7 730
43 210 723 730 832 838 997
```

Output

```
730 730
```

Explanation

730 is present in the array so it's floor and ceil both are 730 and 730.

Example 2

Input

```
10 1
24 50 62 74 87 434 477 625 783 940
```

Output

```
-1 24
```

Explanation

The floor of key 1 is not present in the array,so it's floor is -1 but it's ceiling according to our requirement is 24 which is in the array.

Constraints

1 <= n <= 10^5

1 <= key <= 10^9

1 <= arr[i] <= 10^9

Topic Tags

- **Binary Search**
- **Arrays**

My code

```
// in java
import java.util.*;

public class Main {
     public static int findFloor(int key, int[] arr) {
        int l = 0, r = arr.length - 1;
        int res = -1;
        while (l <= r) {
            int m = l + (r - l) / 2;
            if (arr[m] == key)
                return key;
            if (arr[m] < key) {
                res = arr[m];
                l = m + 1;
            } else {
                r = m - 1;
            }
        }
        return res;
    }

    public static int findCeil(int key, int[] arr) {
        int l = 0, r = arr.length - 1;
        int res = -1;
        while (l <= r) {
            int m = l + (r - l) / 2;
            if (arr[m] == key)
```

```
            return key;
        if (arr[m] > key) {
            res = arr[m];
            r = m - 1;
        } else {
            l = m + 1;
        }
    }
    return res;
}

public static int[] floorAndCeil(int key, int[] arr) {
    int[] result = new int[2];
    result[0] = findFloor(key, arr);
    result[1] = findCeil(key, arr);
    return result;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n, key;
    n = sc.nextInt();
    key = sc.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++)
        arr[i] = sc.nextInt();
    int[] result = floorAndCeil(key, arr);
    System.out.println(result[0] + " " + result[1]);
    sc.close();
}
```

```
}
/*import java.util.*;

public class Main {
    public static int[] floorAndCeil(int k, int[] arr) {
        //Write code here
              int ar[]=new int[2];
                int lp=0,rp=arr.length;
           while(lp<=rp)
                  {
                        int mid=(lp+rp)/2;
                        if(arr[mid]==k)
                        {
                              // System.out.print(mid);
                                                ar[0]=arr[mid];
                                                ar[1]=arr[mid];
                               return ar;
                        }
                        else if(arr[mid]>k)
                               rp=mid-1;
                        else lp=mid+1;

                  }
            // System.out.print(lp-1);
                              if(lp==0)   ar[0]=-1;
                             else     ar[0]=arr[lp-1];

                           if(lp==arr.length)
                                                ar[1]=-1;
                                                else
```

```
                                        {
                                            while(arr[lp]==arr[lp+1])
                                                lp=lp+1;
                                            ar[1]=arr[lp];
                                        }



                                return ar;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n, key;
        n = sc.nextInt();
        key = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt();
        int[] result = floorAndCeil(key, arr);
        System.out.println(result[0] + " " + result[1]);
        sc.close();
    }
}*/
```