4.c. Finding Floor Value

Aim: Given a sorted array and a value x, the floor of x is the largest element in array smaller than or equal to x. Write divide and conquer algorithm to find floor of x.

Input Format

First Line Contains Integer n – Size of array
Next n lines Contains n numbers – Elements of an array
Last Line Contains Integer x – Value for x

Output Format

First Line Contains Integer – Floor value for x

Algorithm:

```
int large(arr, l, r, x){
    // Base case: if the range is invalid
    if r < l
        return 0  // return 0 when there is no valid element

    // Calculate the middle index
    mid = (l + r) / 2

    // Check if the middle element is equal to x
    if arr[mid] is equal to x
        return mid // return the index of x if found

    // If the middle element is less than x
    else if arr[mid] < x
        // Recursively search in the right half
        floorIndex = large(arr, mid + 1, r, x)

        // Check if a valid floor index is found
```

```
        if floorIndex is not equal to 0
            return floorIndex // return the found index
        else
            return mid // return mid as the largest element less than x

    // If the middle element is greater than x, search in the left half
    else
        return large(arr, l, mid - 1, x) // search in the left half
}
Int main()
    initialize n // number of elements in the array
    read n from user

    initialize arr of size n // array to hold input values

    // Read values into the array
    for i from 0 to n - 1
        read arr[i] from user

    initialize l as 0 // left index
    initialize r as n - 1 // right index

    initialize x // the value for which we want to find the largest element less than or equal to
x
    read x from user

    // Call the large function
    result = large(arr, l, r, x)
```

```
    // Check the result if
    result is equal to 0
        print x // if no valid element, print x
    else
        print arr[result] // print the largest element less than or equal to x
```

Program:

```
#include<stdio.h>
int large(int arr[],int l,int r,int x){
    if (r < l) {
        return 0;
    }
    int mid=(l+r)/2;
    if (arr[mid]==x)
    {
        return mid;
    }
    else if (arr[mid]<x)
    {
        int floorIndex=large(arr,mid+1,r,x);
        if(floorIndex!=0)
        {
            return floorIndex;
        }
        else
        {
           return floorIndex=mid;
```

```
        }
    }
    else
    {
        return large(arr,l,mid-1,x);
    }
}

int main(){
    int n;
    scanf("%d",&n);
    int arr[n];
    for (int i=0;i<n;i++){ scanf("%d
        ",&arr[i]);
    }
    int l=0;
    int r=n-1;
    int x;
    scanf("%d",&x);
    int result=large(arr,l,r,x);
    if (result == 0)
    {
        printf( "%d",x);
    }
    else
    {
        printf( "%d",arr[result]);
    }
```

}

Output:

	Input	Expected	Got	
✔	6 1 2 8 10 12 19 5	2	2	✔
✔	5 10 22 85 108 129 100	85	85	✔
✔	7 3 5 7 9 11 13 15 10	9	9	✔