DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SUBJECT CODE:
DESIGN AND ANALYSIS OF ALGORITHMS

Lab 1:

Date of the Session: ___/___/___ **Time of the Session: _____to_____**

Pre-Lab Task:

1. Calculate the time complexity of the following:

```
int binarySearch(int a, int low, int high, int tar)
{
int mid;
if (low > high)
return (0);
mid = floor((low + high)/2)
if (a[mid] == tar)
        return (mid)
else
        if tar < a[mid]
                return binarySearch(a, low, mid-1, tar)
        else
                return binarySearch(a, mid+1, high, tar)
 }
```

2.
```
function fn(n)
{
if(n<0) return 0;
if(n<2) return n;
return fn(n-1)+fn(n-2);
}
```

3) The median of a list of numbers is essentially its middle element after sorting. The same number of elements occur after it as before. Given a list of numbers with an odd number of elements, find the median?

Example

arr=[5,3,1,2,4]

The sorted array a'=[1,2,3,4,5]. The middle element and the median is 3.

Function Description

Complete the *findMedian* function in the editor below.

findMedian has the following parameter(s):

- *int arr[n]:* an unsorted array of integers

Returns

- *int:* the median of the array

Sample Input 0

7
0 1 2 4 6 5 3

Sample Output 0

3

In-Lab Task:

1) Given an array of strings **arr[]**, the task is to sort the array of strings according to frequency of each string, in ascending order. If two elements have same frequency, then they are to be sorted in alphabetical order.

   Input: arr[] = {“Ramesh”, “Mahesh”, “Mahesh”, “Ramesh”}

   Output: (“Mahesh”,”Ramesh”}

   Explanation:
   As both the strings have same frequency, the array is sorted in the alphabetical order.

2) Given an array of strings **words[]** and the **sequential order** of alphabets, our task is to sort the array according to the order given. Assume that the dictionary and the words only contain lowercase alphabets.

Input: words = {“word”, “world”, “row”}, order = “worldabcefghijkmnpqstuvxyz”

Output: “world”,“word”,“row”

Explanation:
According to the given order ‘l’ occurs before ‘d’ hence the words “world” will be kept first.

Post-Lab Task:

1) Given an array arr[] of N strings, the task is to sort these strings according to the number of upper-case letters in them try to use zip function to get the format.

   Input arr[] = {poiNtEr, aRRAy, cOde, foR}

   Output: [('cOde', 1), ('foR', 1), ('poiNtEr', 2), ('aRRAy', 3)]

   “aRRAy” R, R, A->3 Upper Case Letters

   “poiNtEr” N, E->2 Upper Case Letters

   “cOde” O->2 Upper Case Letters

   “foR” R->3 Upper Case Letters

2) In KLU streets we have lots of electrical poles.
Note: all poles are sorted with respective to their heights.

Professor Hari Vege given the H = height of one pole to Mothi then asked him to print the position of that pole, here we consider index as a position. Mothi is particularly good at algorithms, so he written an algorithm to find that position. But he is extremely poor at finding time complexity. Your task is to help your friend Mothi to analyze the time complexity of the given problem.

```
Int BinarySearch (int a, int low, int high, int tar)
{
    int mid;
    if (low > high) return 0;
    mid = floor((low + high)/2)
    if (a[mid] == tar)
            return mid;
    else if (tar < a[mid])
            return BinarySearch (a, low, mid-1, tar)
    else
            return BinarySearch (a, mid+1, high, tar)
}
```

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured: _____ out of _____ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: