Ex. No: 4 **Date: 03.09.24**

Register No.: 230701520 **Name: S Raghavan**

Divide and Conquer

4.a. Number of Zeros in a Given Array

Aim: Given an array of 1s and 0s this has all 1s first followed by all 0s. Aim is to find the number of 0s. Write a program using Divide and Conquer to Count the number of zeroes in the given array.

Input Format

First Line Contains Integer m – Size of array

Next m lines Contains m numbers – Elements of an array

Output Format

First Line Contains Integer – Number of zeroes present in the given array.

Algorithm:

```
function count(a, left, right) {
    // base case: if left index exceeds right index
    if left is greater than right {
        return 0
    }

    initialize mid as (left + right) / 2 // find the middle index

    // check if the middle element is 1
    if a[mid] is equal to 1 {
        // check if the next element is 0
        if a[mid + 1] is equal to 0 {
            // count zeros from mid + 1 to right
```

```
            initialize c as (right - (mid + 1)) + 1
            return c
        } else {
            // search in the right half return
            count(a, mid + 1, right)
        }
    }
    // check if both ends are 0
    else if a[left] is equal to 0 and a[right] is equal to 0 {
        return right + 1 // return total count of elements
    }
    // search in the left half
    else {
        return count(a, left, mid - 1)
    }
}

function main() {
    initialize n // number of elements
    read n from user

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

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

initialize left as 0 // left index

initialize right as n – 1 // ri

Program:

```
#include <stdio.h>
int count(int a[],int left,int right)
{
    if(left>right)
    {
        return 0;
    }
    int mid=(left+right)/2;
    if(a[mid]==1)
    {
        if(a[mid+1]==0)
        {
            int c= (right-(mid+1))+1;
            return c;
        }
        else{
            return count(a,mid+1,right);
        }
    }
    else if(a[left]==0 && a[right]==0)
    {
        return right+1;
    }
    else
```

```
    {
        return count(a,left,mid-1);
    }

}
int main()
{
    int n;
    scanf("%d",&n);
    int arr[n];
    for(int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    int left=0;
    int right=n-1;
    int result=count(arr,left,right);
    printf("%d",result);
}
```

Output:

	Input	Expected	Got	
✔	5 1 1 1 0 0	2	2	✔
✔	10 1 1 1 1 1 1 1 1 1 1	0	0	✔
✔	8 0 0 0 0 0 0 0 0	8	8	✔