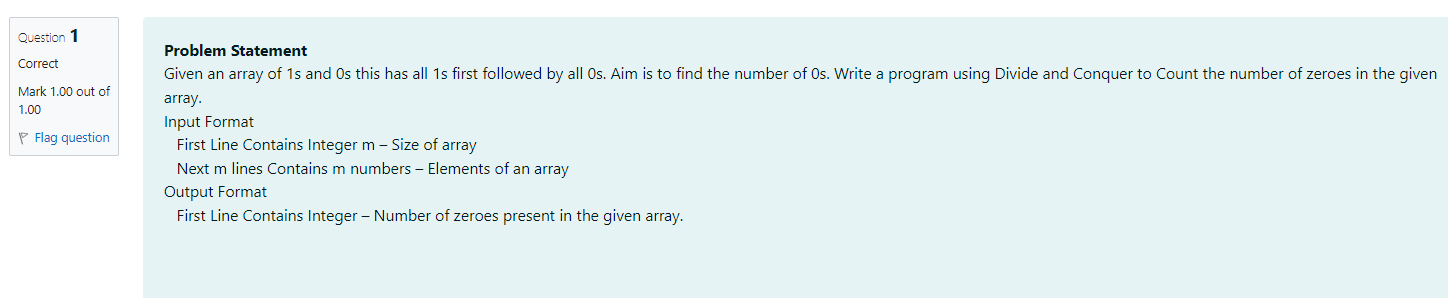
[**Divide and Conquer**](http://118.185.187.137/moodle/course/view.php?id=155#section-4)

**NAME:B.Elumalai**

**ROLL No:230701084**

**1-Number of Zeros in a Given Array**

****

**CODE**

#include<stdio.h>

int main(){

int m,c=0;

scanf("%d",&m);

int b=m/2;

int a[m];

for(int i=0;i<m;i++)

{

scanf("%d",&a[i]);

}

{

for(int i=0;i<b;i++)

{

if(a[i]==0)

{

c=c+1;

}

}

for(int j=b;j<m;j++){

if(a[j]==0){

c=c+1;

}

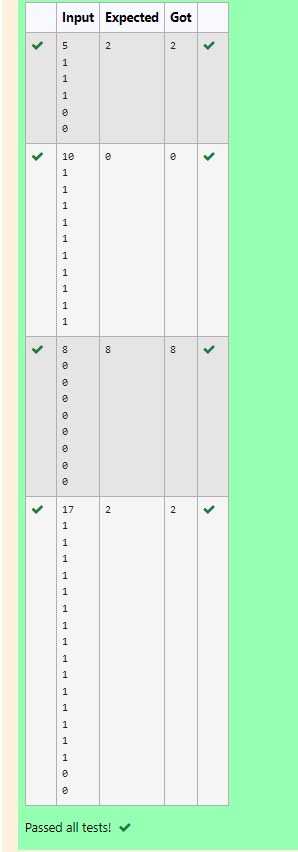
}

printf("%d",c);

}

}

**OUTPUT**

****

## 2-Majority Element

## 

## CODE

## #include <stdio.h>

## int c=0;

## int C(int arr[],int x,int y,int k)

## {

## int m=x+(y-x)/2;

## if (arr[m]==k)

## c++;

## else

## {

## C(arr,x,m,k);

## C(arr,m+1,y,k);

## }

## return c;

## 

## }

## int main()

## {

## int a;

## scanf("%d",&a);

## int arr[a];

## for (int i=0;i<a;i++)

## scanf("%d",&arr[i]);

## int b=arr[0];

## if (C(arr,0,a,b)>a/2)

## printf("%d",b);

## else

## {

## for (int i=0;i<a/2;i++)

## if (arr[i]!=b)

## {

## printf("%d",b);

## break;

## }

## }

## 

## }

## OUTPUT

## 

## 3-Finding Floor Value

## 

## CODE

## #include<stdio.h>

## int f(int arr[],int a,int b)

## {

## int l=0, h=a-1;

## int fv=-1;

## while(l<=h)

## {

## int m=(l+h)/2;

## if (arr[m]<=b)

## {

## fv=arr[m];

## l=m+1;

## }

## else{

## h=m-1;

## }

## }

## return fv;

## }

## int main()

## {

## int a,b;

## scanf("%d",&a);

## int(arr[a]);

## for(int i=0;i<a;i++)

## {

## scanf("%d",&arr[i]);

## }

## scanf("%d",&b);

## int v=f(arr,a,b);

## printf("%d",v);

## }

## OUTPUT

## 

## 4-Two Elements sum to x

## 

## CODE

## #include <stdio.h>

## int main()

## {

## int n;

## scanf("%d",&n);

## int arr[n];

## for(int i=0;i<n;i++)

## {

## scanf("%d",&arr[i]);

## }

## 

## int sum;

## scanf("%d",&sum);

## for(int i=0;i<n;i++)

## {

## int m= arr[i]+arr[i+n];

## 

## if(m==sum)

## {

## printf("%d\n",arr[i+n]);

## printf("%d",arr[i]);

## break;

## }

## if (i==n-1 )

## printf("No");

## }

## 

## }

## OUTPUT

## 

## 5-Implementation of Quick Sort

## 

## CODE

## #include <stdio.h>

## void quick\_sort(int arr[], int low, int high) {

## if (low < high) {

## int pivot = arr[high];

## int i = low - 1;

## for (int j = low; j < high; j++) {

## if (arr[j] < pivot) {

## i++;

## int temp = arr[i];

## arr[i] = arr[j];

## arr[j] = temp;

## }

## }

## int temp = arr[i + 1];

## arr[i + 1] = arr[high];

## arr[high] = temp;

## quick\_sort(arr, low, i);

## quick\_sort(arr, i + 2, high);

## }

## }

## int main() {

## int n;

## scanf("%d", &n);

## int arr[n];

## for (int i = 0; i < n; i++) {

## scanf("%d", &arr[i]);

## }

## quick\_sort(arr, 0, n - 1);

## for (int i = 0; i < n; i++) {

## printf("%d ", arr[i]);

## }

## printf("\n");

## return 0;

## }

## OUTPUT

## 