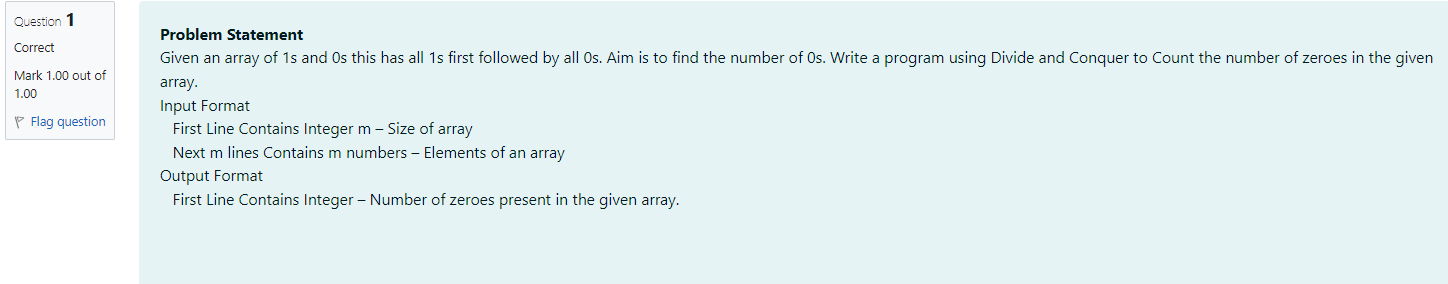
DIVIDE AND CONQUER

**NAME:ENIYA.B.A**

**ROLL NO:230701085**

1-Number of Zeros in a Given Array

****

**CODE:**

#include<stdio.h>

int count(int a[], int l, int h){

if(l>h){

return 0;

}

if(l==h){

if(a[l]==0){

return 1;

}

else{

return 0;

}

}

int mid = (l+h)/2;

int lm = count(a, l, mid);

int rm = count(a, mid+1, h);

return rm+lm;

}

int main(){

int m;

scanf("%d",&m);

int a[m];

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

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

}

int ans = count(a, 0, m-1);

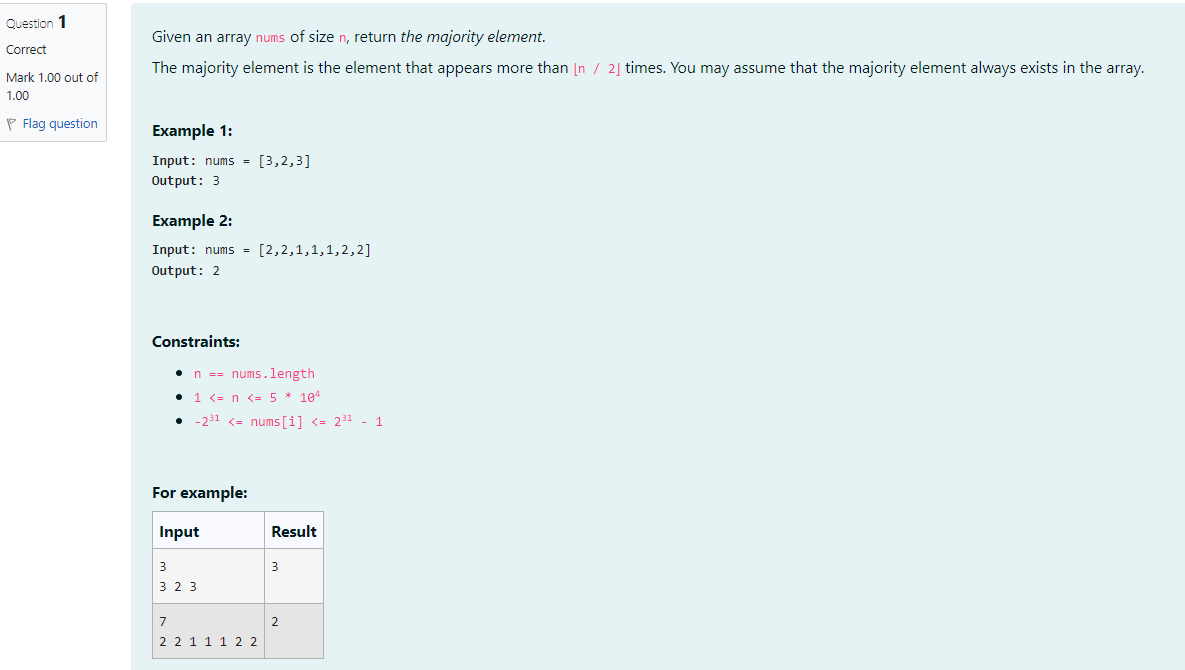
printf("%d",ans);

}

**OUTPUT:**



2-Majority Element



**CODE:**

#include<stdio.h>

int divide(int a[],int l,int r,int s)

{

if(l==r)

{

return a[l];

}

int mid=(l+r)/2;

int left=divide(a,l,mid,s);

int right=divide(a,mid+1,r,s);

int lcount=0,rcount=0;

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

{

if(a[i]==left)

lcount++;

if(a[i]==right)

rcount++;

}

if(lcount>(s/2))

return left;

else

return right;

}

int main()

{

int size;

scanf("%d",&size);

int arr[size];

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

{

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

}

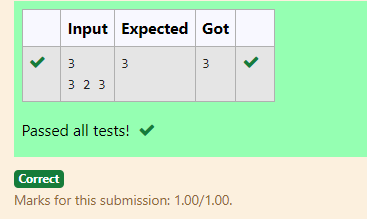
int low=0,high=size-1;

int majority=divide(arr,low,high,size);

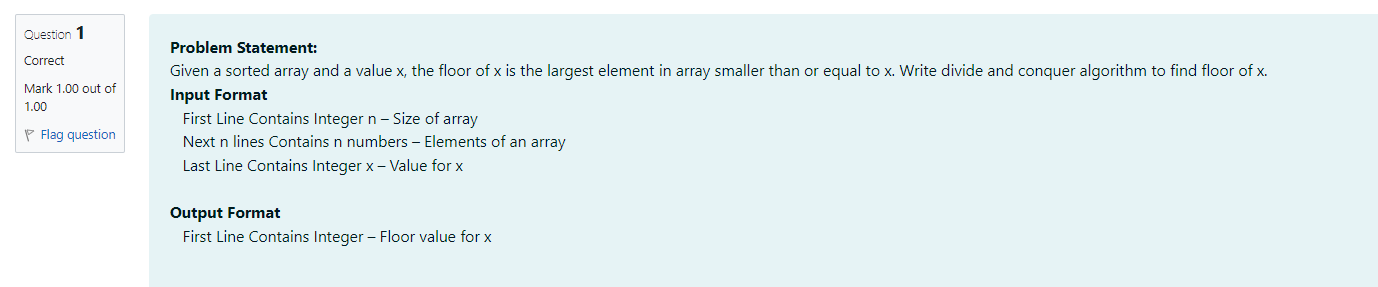
printf("%d",majority);

}

**OUTPUT:**



## 3-Finding Floor Value



**CODE:**

#include <stdio.h>

int Floor(int a[],int n,int x) {

int l=0,h =n-1;

int f=-1;

while(l<=h){

int mid=(l+h)/2;

if (a[mid]==x) return a[mid];

else if (a[mid]<x){

f=a[mid];

l=mid+1;}

else h=mid-1;

}

return f; }

int main(){

int n,x;

scanf("%d",&n);

int a[n];

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

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

scanf("%d", &x);

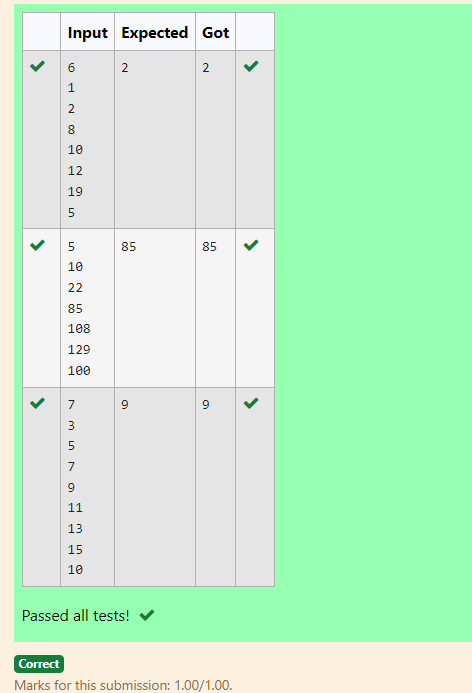
int f= Floor(a,n,x);

printf("%d\n", f);

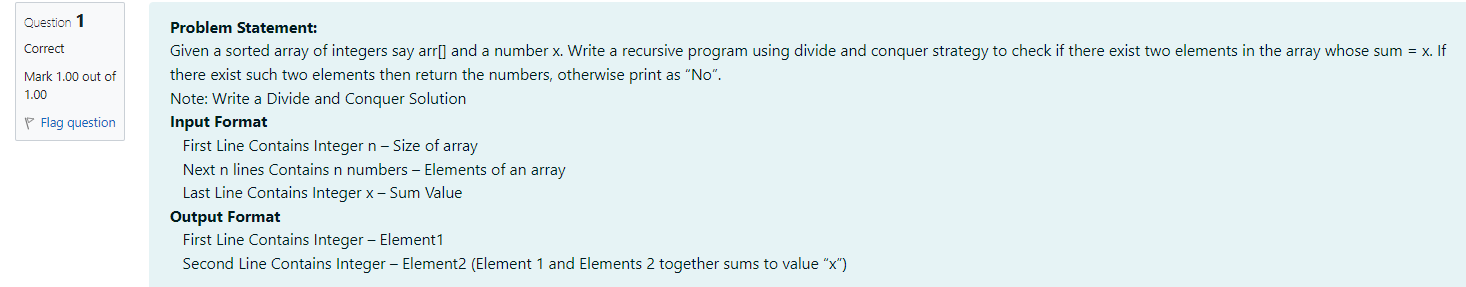
return 0;

}

**OUTPUT:**



4-Two Elements sum to x



**CODE:**

#include<stdio.h>

void sum(int a[],int l,int r,int x){

if (l >= r) {

printf("No\n");

return;

}

int asum=a[l]+a[r];

if(asum==x){

printf("%d\n",a[l]);

printf("%d\n",a[r]);

}else if(asum<x){

sum(a,l+1,r,x);

}else{

sum(a,l,r-1,x);

}

}

int main(){

int n,x;

scanf("%d",&n);

int a[n];

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

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

}

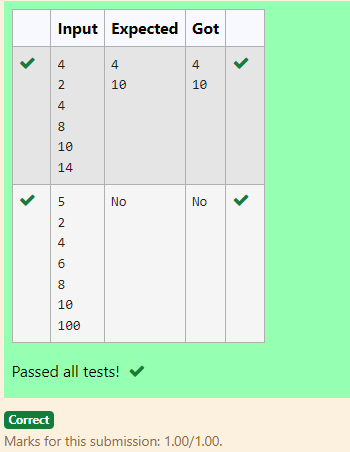
scanf("%d",&x);

sum(a,0,n-1,x);

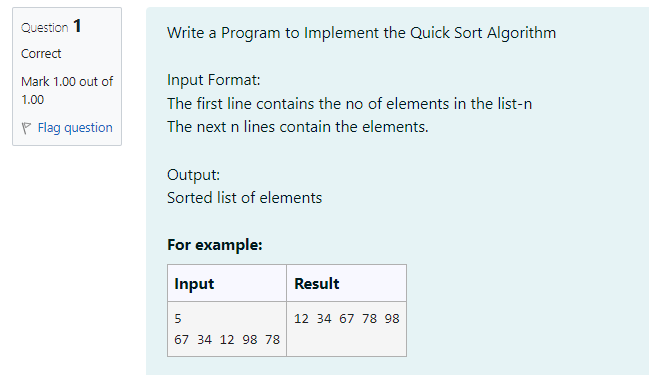
return 0;

}

**OUTPUT:**



## 5-Implementation of Quick Sort



**CODE:**

#include <stdio.h>

void QuickSort(int a[], int left, int right) {

int i, j, temp, pivot;

if (left < right) {

pivot = left;

i = left + 1;

j = right;

while (i <= j) {

while (i <= right && a[i] < a[pivot]) {

i++;

}

while (a[j] > a[pivot]) {

j--;

}

if (i <= j) {

temp = a[i];

a[i] = a[j];

a[j] = temp;

i++;

j--;

}

}

temp = a[pivot];

a[pivot] = a[j];

a[j] = temp;

QuickSort(a, left, j - 1);

QuickSort(a, j + 1, right);

}

}

int main() {

int i, n;

scanf("%d", &n);

int a[n];

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

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

}

QuickSort(a, 0, n - 1);

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

printf("%d ", a[i]);

}

return 0;

}

**OUTPUT:**

