**COMPETITIVE PROGRAMMING**

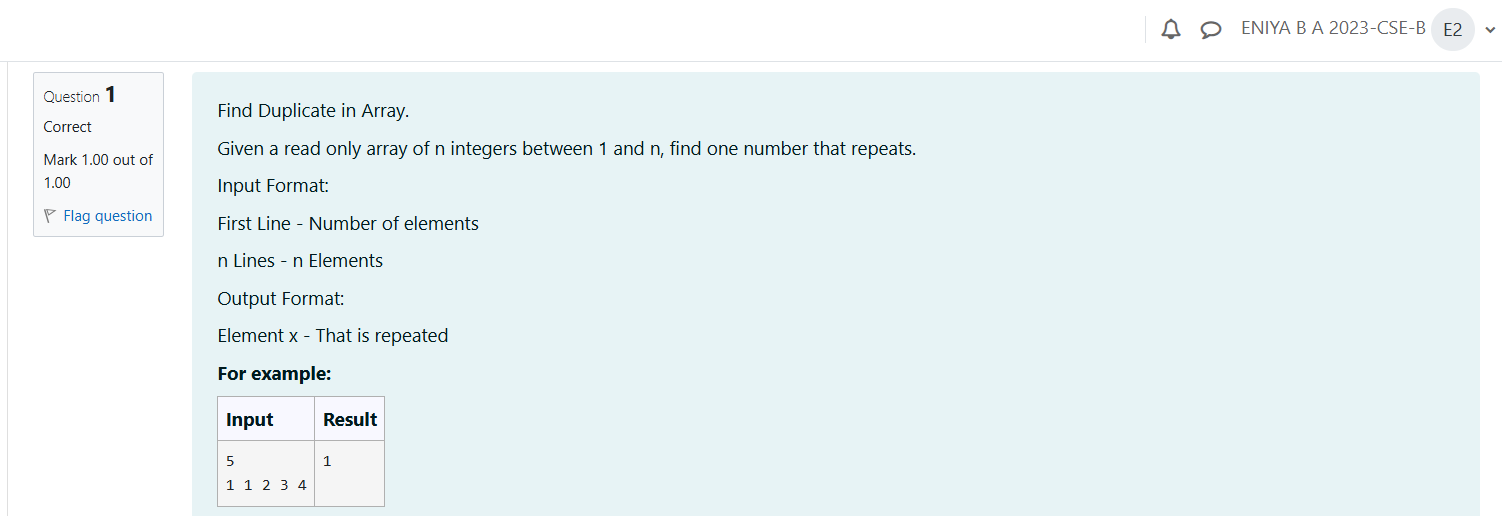
**NAME : ENIYA.B.A**

**CLASS : CSE-B**

**ROLL NO : 230701085**

**SUB : DESIGN AND ANALYSIS OF ALGORITHM**

1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity



**CODE:**

#include<stdio.h>

int main(){

int n,i,j,count=0;

scanf("%d",&n);

int a[n];

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

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

}

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

for(j=i+1;j<n;j++){

if(a[i]==a[j]){

count=a[j];

}

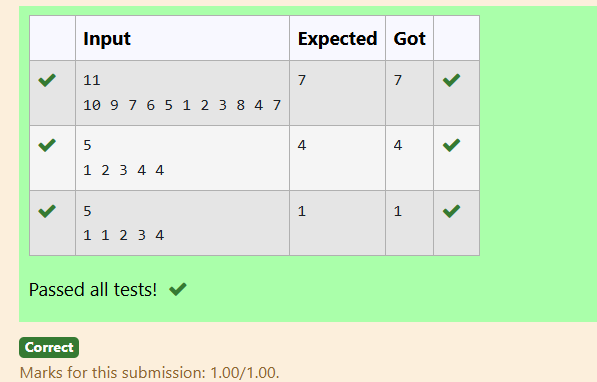
}

}

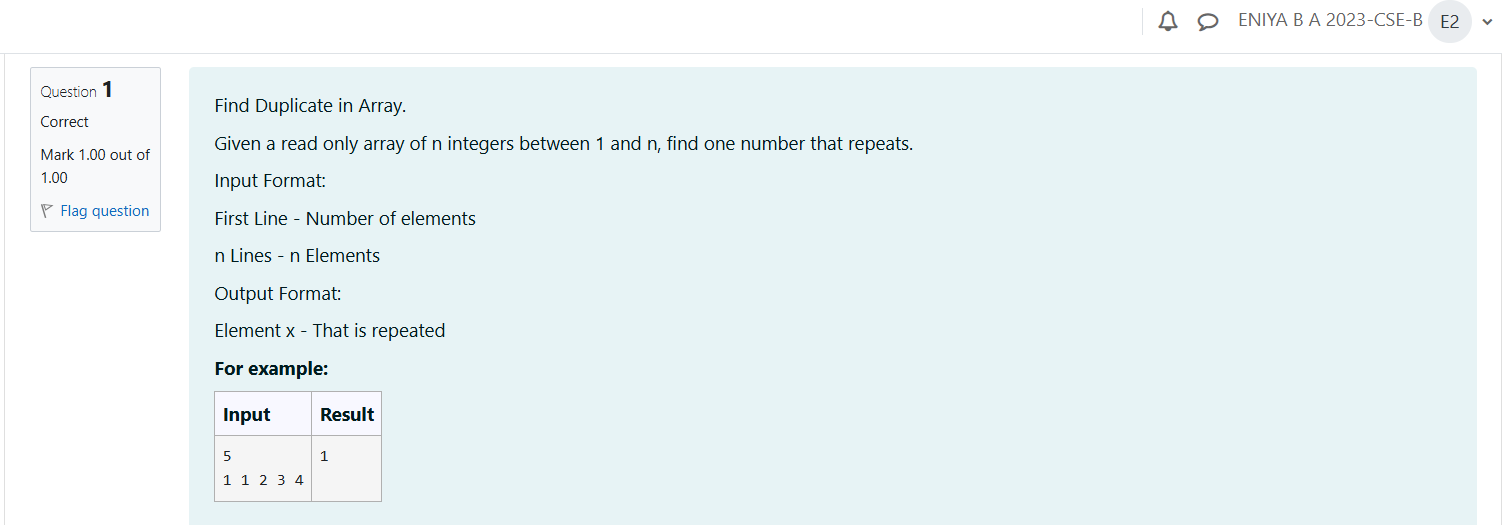
printf("%d",count);

}

**OUTPUT:**



2-Finding Duplicates-O(n) Time Complexity,O(1) Space Complexity



**CODE:**

#include<stdio.h>

int main(){

int n,i;

scanf("%d",&n);

int a[n],count=0;

scanf("%d",&a[0]);

for(i=1;i<n;i++){

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

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

if(a[i]==a[j]){

count=a[i];

}

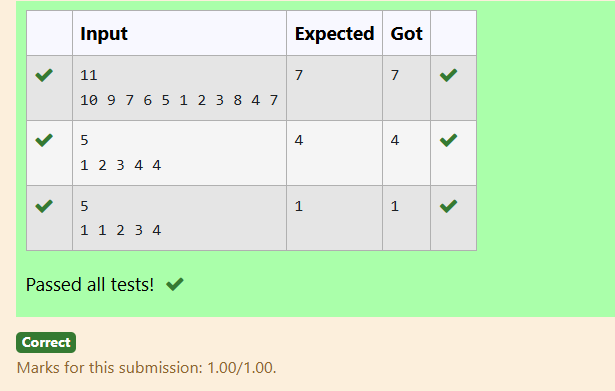
}

}

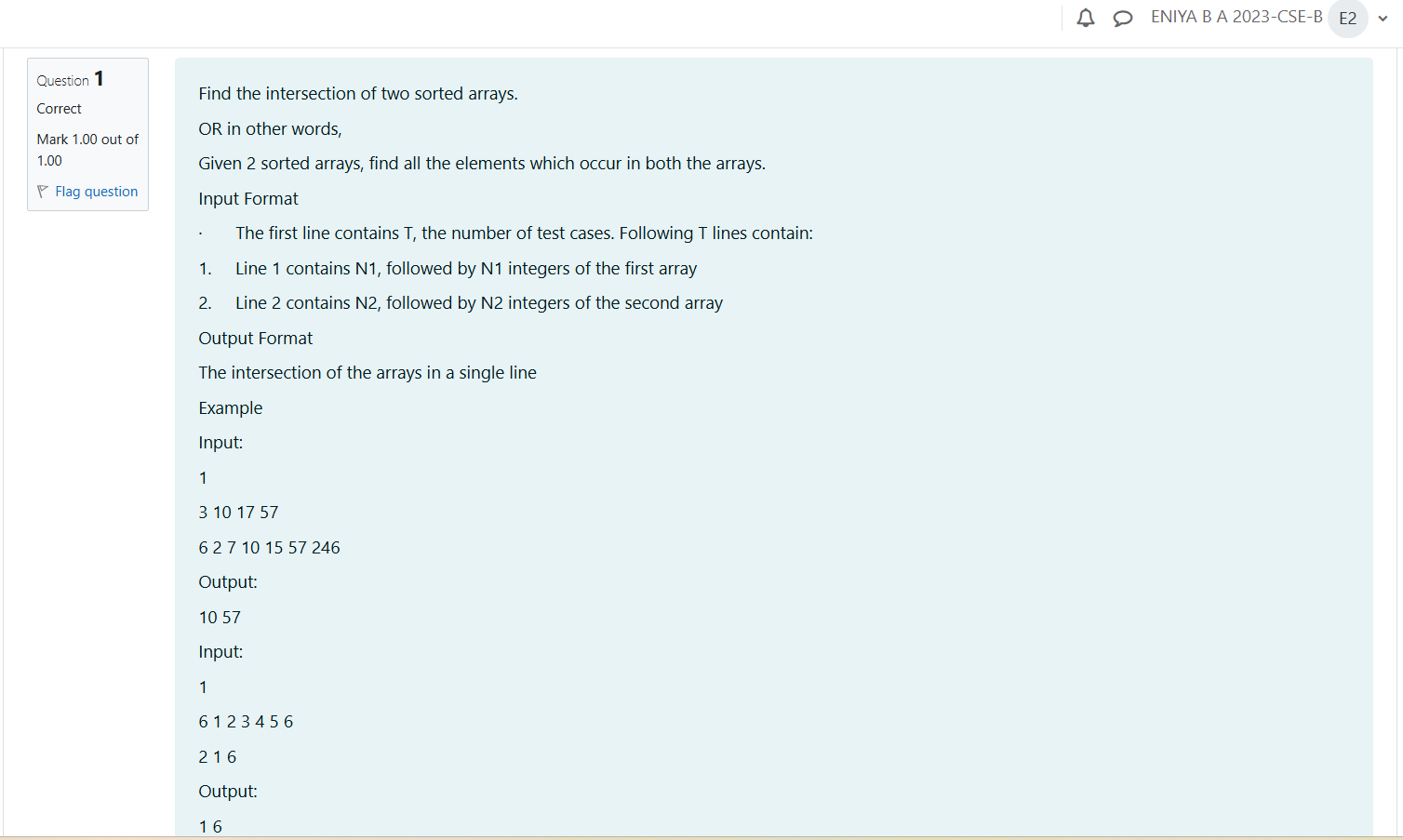
printf("%d",count);

}

**OUTPUT:**



## 3-Print Intersection of 2 sorted arrays-O(m\*n)Time Complexity,O(1) Space Complexity



**CODE:**

#include <stdio.h>

void findIntersection(int arr1[], int n1, int arr2[], int n2) {

int i = 0, j = 0;

int first = 1;

while (i < n1 && j < n2) {

if (arr1[i] < arr2[j]) {

i++;

} else if (arr1[i] > arr2[j]) {

j++;

} else {

if (first) {

printf("%d", arr1[i]);

first = 0;

} else {

printf(" %d", arr1[i]);

}

i++;

j++;

}

}

if (first) {

printf(" ");

}

}

int main() {

int T;

scanf("%d", &T);

while (T--) {

int n1;

scanf("%d", &n1);

int arr1[n1];

for (int i = 0; i < n1; i++) {

scanf("%d", &arr1[i]);

}

int n2;

scanf("%d", &n2);

int arr2[n2];

for (int i = 0; i < n2; i++) {

scanf("%d", &arr2[i]);

}

findIntersection(arr1, n1, arr2, n2);

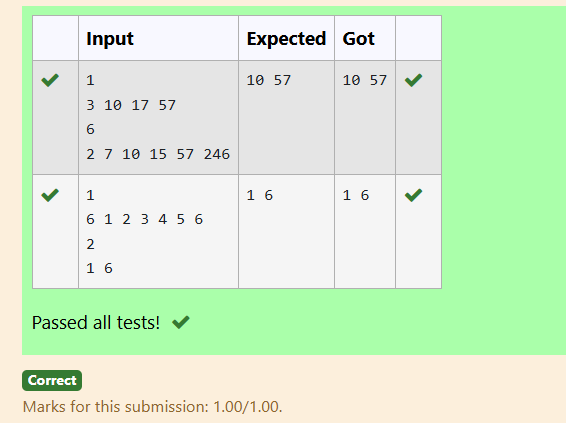
printf("\n");

}

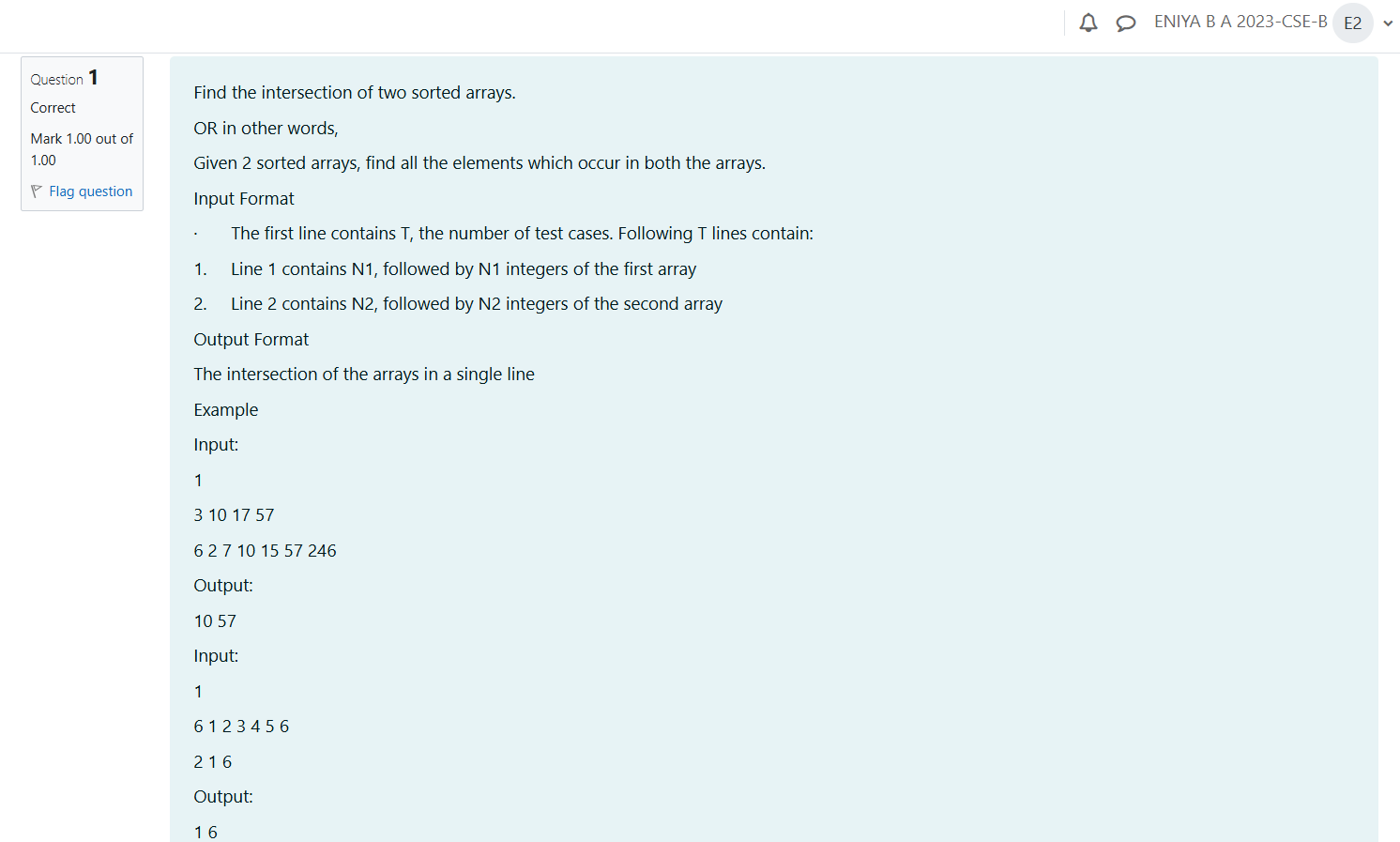
return 0;

}

**OUTPUT:**



## 4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity

****

**CODE:**

#include <stdio.h>

void findIntersection(int arr1[], int n1, int arr2[], int n2) {

int i = 0, j = 0;

int first = 1;

while (i < n1 && j < n2) {

if (arr1[i] < arr2[j]) {

i++;

} else if (arr1[i] > arr2[j]) {

j++;

} else {

if (first) {

printf("%d", arr1[i]);

first = 0;

} else {

printf(" %d", arr1[i]);}

i++;

j++;

}

}

}

int main() {

int T;

scanf("%d", &T);

while (T--) {

int n1;

scanf("%d", &n1);

int arr1[n1];

for (int i = 0; i < n1; i++) {

scanf("%d", &arr1[i]);

}

int n2;

scanf("%d", &n2);

int arr2[n2];

for (int i = 0; i < n2; i++) {

scanf("%d", &arr2[i]);

}

findIntersection(arr1, n1, arr2, n2);

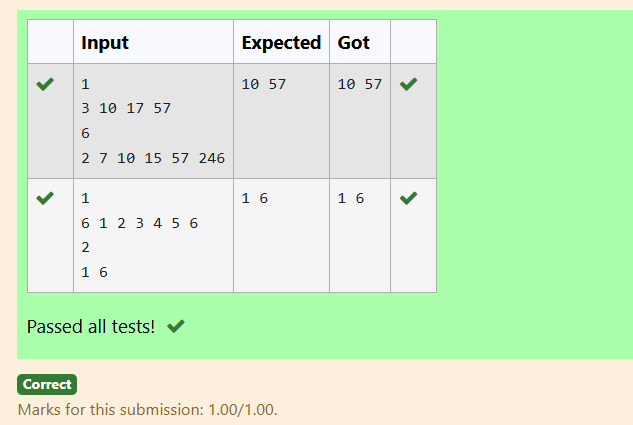
printf("\n");

}

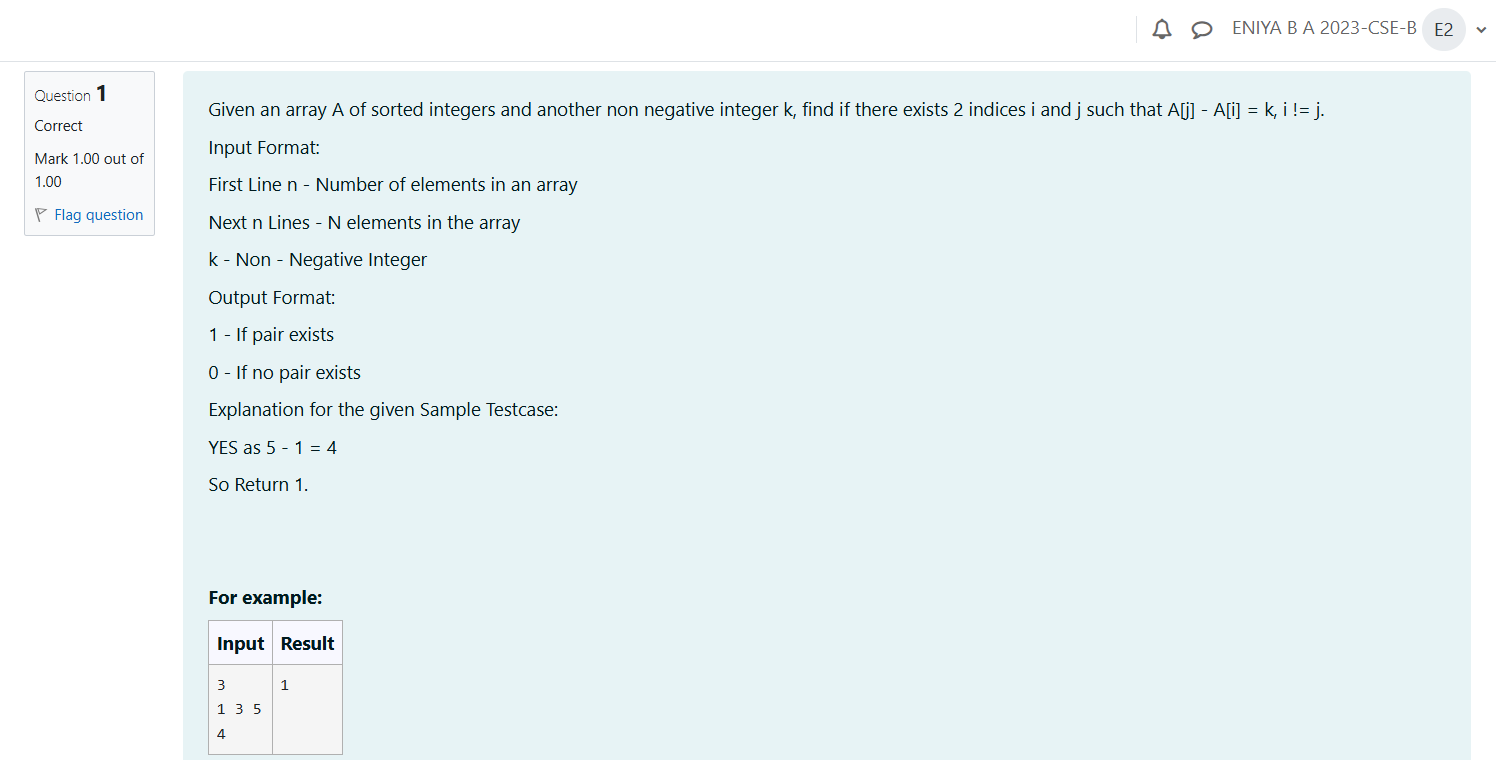
return 0;

}

**OUTPUT:**



## 5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Complexity



**CODE:**

#include<stdio.h>

#include<stdlib.h>

void array(int n,int a[],int k){

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

for(int j=i+1;j<n;j++){

if(abs(a[i]-a[j])==k){

printf("1");

return;

}

}

}

printf("0");

}

int main(){

int n,k;

scanf("%d",&n);

int a[n];

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

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

}

scanf("%d",&k);

array(n,a,k);

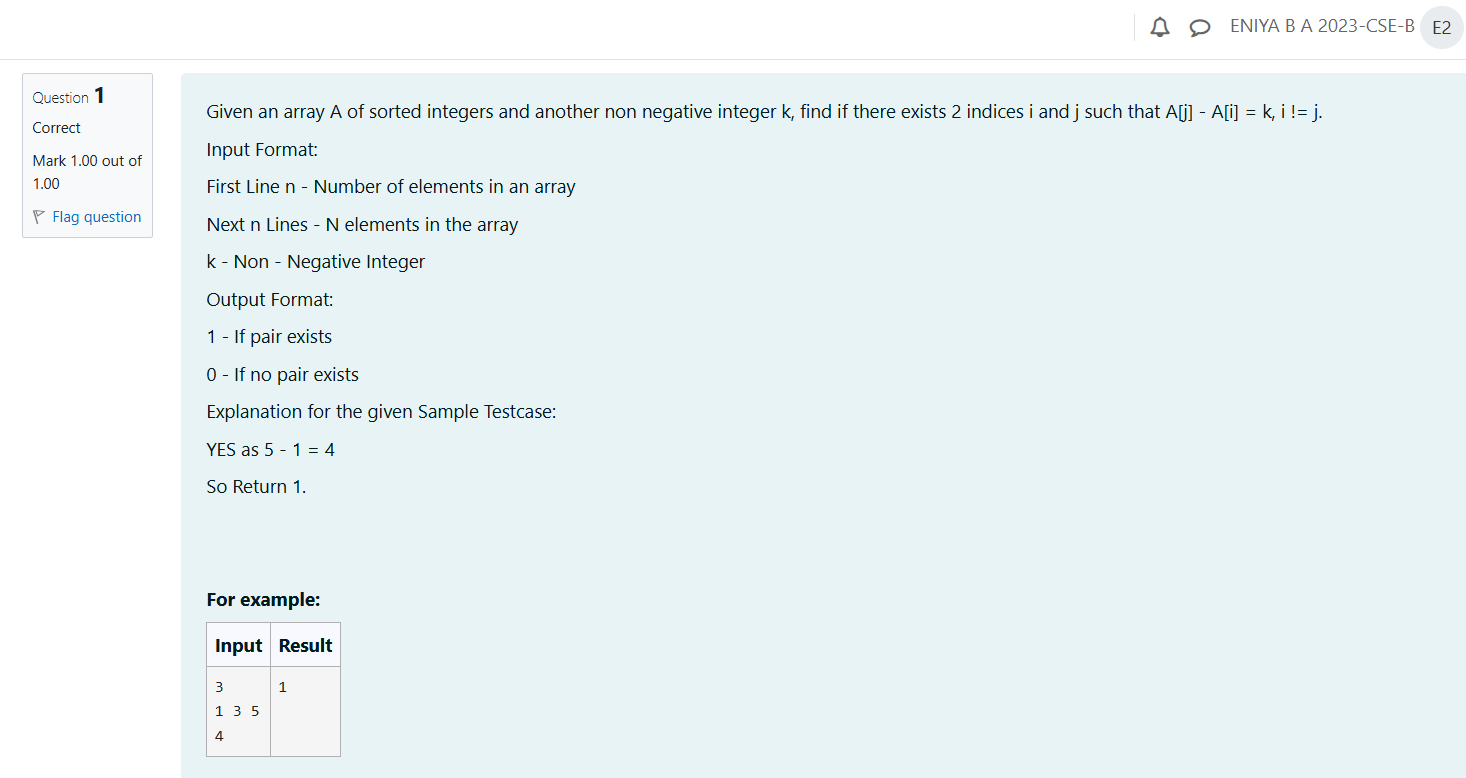
return 0;

}

**OUTPUT**:



## 6-Pair with Difference -O(n) Time Complexity,O(1) Space Complexity



**CODE:**

#include <stdio.h>

int main() {

int n, k;

scanf("%d", &n);

int a[n];

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

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

}

scanf("%d", &k);

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

for (int j = i + 1; j < n; j++) {

if (a[j] - a[i] == k) {

printf("1\n");

return 0;}

}

}

printf("0\n");

return 0;

}

**OUTPUT:**

