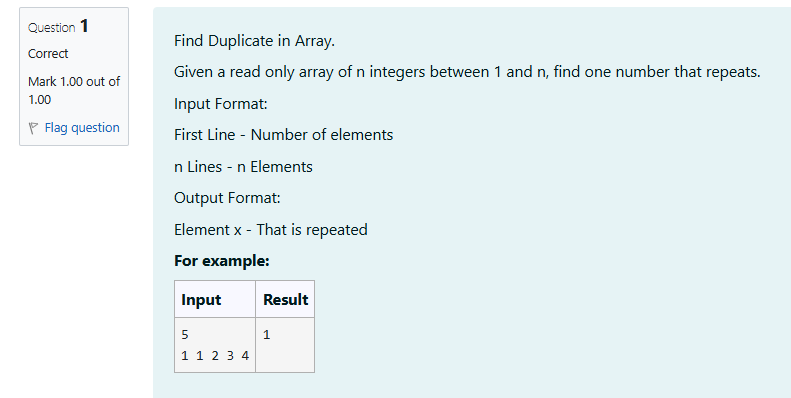
**COMPETITIVE PROGRAMMING**

**NAME: P JAYAPRADHA**

**ROLL NO: 230701130**

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

****

**CODE**

#include<stdio.h>

int main()

{

int n,c=0;

scanf("%d",&n);

int a[n];

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

{

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

}

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

{

for(int j=0;j<n;j++)

{

if (i!=j&&a[i]==a[j])

{

c=c+a[i];

break;

}

}

if(c>0)

{

break;

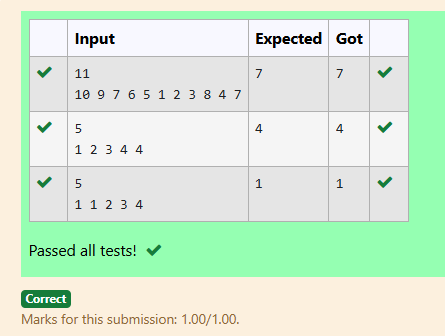
}

}

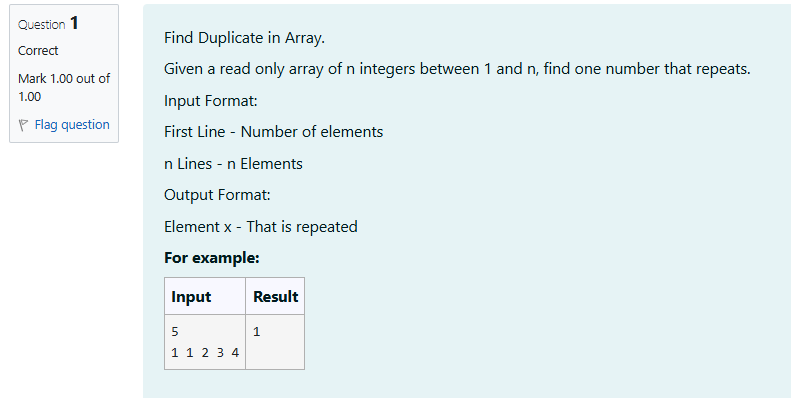
printf("%d",c);

}

**OUTPUT**

****

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

****

**CODE**

#include<stdio.h>

int main()

{

int n;

scanf("%d",&n);

int a[n];

int ind,b;

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

{

scanf("%d",&b);

ind=b%n;

if(a[ind]!=0 && a[ind]==b) {

printf("%d",b);

break;

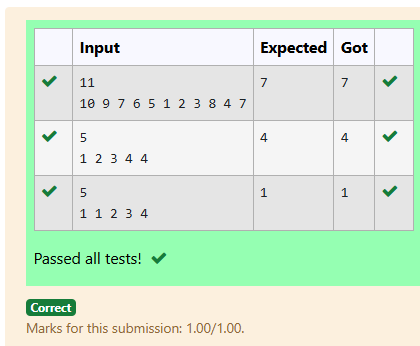
}

a[ind]=b;

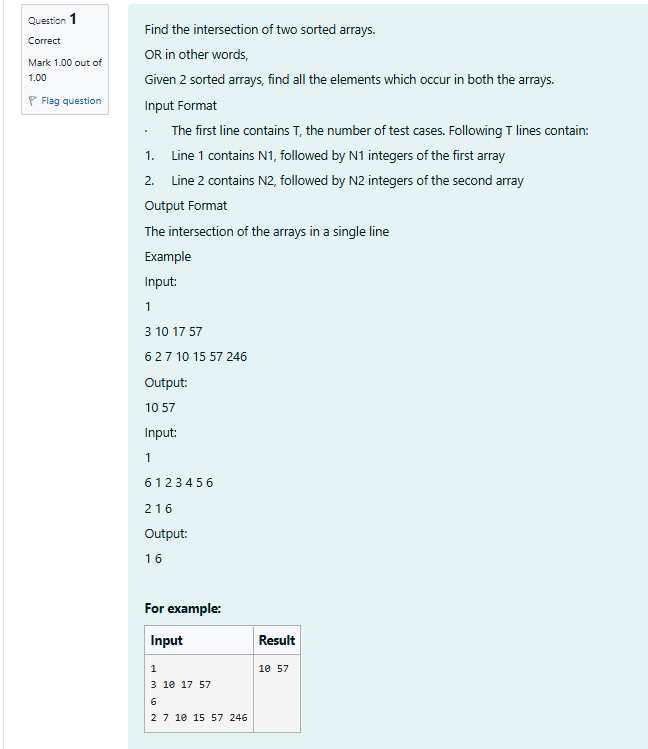
}

}

**OUTPUT**

****

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

****

**CODE**

#include <stdio.h>

void find\_intersection(int arr1[], int n1, int arr2[], int n2) {

int i = 0, j = 0;

int found = 0;

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

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

i++;

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

j++;

} else {

if (found == 0) {

found = 1;

} else {

printf(" ");

}

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

i++;

j++;

}

}

if (found == 0) {

printf(" ");

}

printf("\n");

}

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]);

}

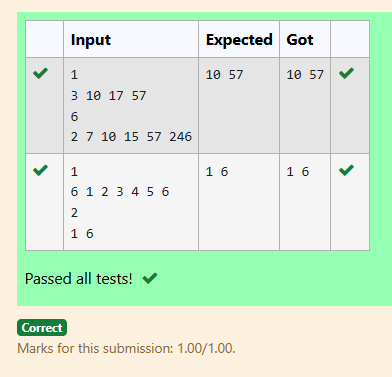
find\_intersection(arr1, n1, arr2, n2);

}

return 0;

}

**OUTPUT**

****

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

## C:\Users\DELL\Pictures\Screenshots\Screenshot 2024-11-05 140919.png

## CODE

#include <stdio.h>

void find\_intersection(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(" ");

}

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

first = 0;

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]);

}

find\_intersection(arr1, n1, arr2, n2);

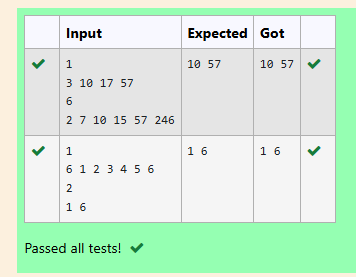
printf("\n");

}

return 0;

}

**OUTPUT**

****

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

**CODE**

#include <stdio.h>

int has\_pair\_with\_difference(int A[], int n, int k) {

int i = 0, j = 0;

while (j < n) {

int diff = A[j] - A[i];

if (diff == k && i != j) {

return 1;

} else if (diff < k) {

j++;

} else {

i++;

if (i == j) {

j++;

}

}

}

return 0;

}

int main() {

int n;

scanf("%d", &n);

int A[n];

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

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

}

int k;

scanf("%d", &k);

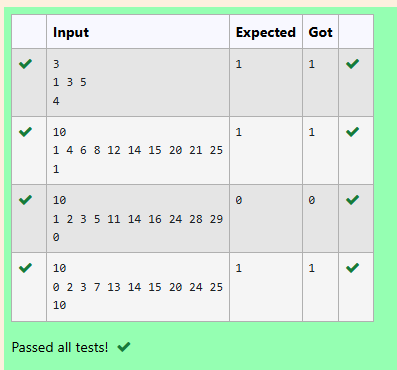
int result = has\_pair\_with\_difference(A, n, k);

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

return 0;

}

**OUTPUT**

****

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

## C:\Users\DELL\Pictures\Screenshots\Screenshot 2024-11-05 141740.png

## CODE

## #include <stdio.h>

## int has\_pair\_with\_difference(int A[], int n, int k) {

## int i = 0, j = 0;

## while (j < n) {

## int diff = A[j] - A[i];

## if (diff == k && i != j) {

## return 1;

## }

## if (diff < k) {

## j++;

## } else {

## i++;

## if (i == j) {

## j++;

## }

## }

## }

## return 0;

## }

## int main() {

## int n;

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

## int A[n];

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

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

## }

## int k;

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

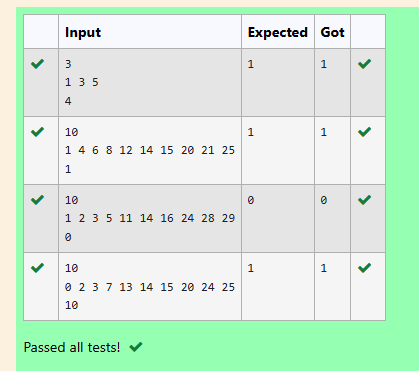
## int result = has\_pair\_with\_difference(A, n, k);

## printf("%d\n", result);

## return 0;

## }

## OUTPUT

****