REG NO: 230701017

NAME : Ajay Srinivas R

DEPT : CSE - A

COMPETITIVE PROGRAMMING

QUESTION 6.A AIM:

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

ALGORITHM:

Step 1: Start

Step 2: Input the integer n, the number of elements in the array.

Step 3: Input n integers into an array a.

Step 4: Initialize r as -1 to store the repeated element.

Step 5: Use a nested loop to check if any element a[i] matches with subsequent elements a[j].

Step 6: If a match is found, set r to the repeated element.

Step 7: If a repeated element is found (r != -1), print the repeated element. **Step 8: Stop**

PROGRAM:

```c
#include <stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++)
    {
       scanf("%d",&a[i]);
    }
    int r=-1;
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(a[i]==a[j])
            {
                r=a[i];
            }
        }
        if (r!=-1){
            break;
        }
    }
    if(r!= -1){
        printf("%d",r);
    }

}
```

OUTPUT:

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

RESULT:

The above progeam is executed successfully .

QUESTION 6.B AIM:

Find Duplicate in Array.

Given a read only array of n integers between 1 and n, find one number that repeats.

Input Format:

First Line - Number of elements

n Lines - n Elements

Output Format:

Element x - That is repeated

For example:

Input	Result
5 1 1 2 3 4	1

ALGORITHM:

Step 1: Start

Step 2: Input the integer n, the number of elements in the array.

Step 3: Input n integers into an array a.

Step 4: Initialize a boolean array r[100] to track whether a number has already been encountered.

Step 5: Iterate through the array a. For each element, check if it has already been seen.

Step 6: If the element is already seen, print it. If not, mark it as seen in r. **Step 7:**

Stop

PROGRAM:

```c
#include <stdio.h>
#include <stdbool.h>

int main() {
    int n;
    scanf("%d", &n);
    int a[n];
    bool r[100] = {false};
    for (int i = 0; i < n; i++) {
        scanf("%d", &a[i]);
        if (r[a[i]]) {
            printf("%d ", a[i]);
        } else {
            r[a[i]] = true;
        }
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	11 10 9 7 6 5 1 2 3 8 4 7	7	7	✔
✔	5 1 2 3 4 4	4	4	✔
✔	5 1 1 2 3 4	1	1	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

Question 6.C AIM:

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

ALLGORITHM:

Step 1: Start

Step 2: Input the number of test cases t.

Step 3: For each test case, input the size n1 of the first array and input the array arr1.

Step 4: Input the size n2 of the second array and input the array arr2.

Step 5: For each element of arr1, check if it exists in arr2.

Step 6: If a match is found, print the element as part of the intersection.

Step 7: Stop

PROGRAM:

```c
#include <stdio.h>
void intersection(int arr1[],int n1,int arr2[],int n2){
    for (int i=0;i<n1;i++){
        int element=arr1[i];
        for (int j=0;j<n2;j++){
            if (arr2[j]==element) {
                printf("%d ",element);
                break;
            }
        }
    }
    printf("\n");
}

int main(){
    int t;
    scanf("%d",&t);
    while(t--){
        int n1,n2;
        scanf("%d",&n1);
        int arr1[n1];
        for(int i=0;i<n1;i++){
            scanf("%d",&arr1[i]);
        }
        scanf("%d",&n2);
        int arr2[n2];
        for(int i=0;i<n2;i++){
            scanf("%d",&arr2[i]);
        }
        intersection(arr1,n1,arr2,n2);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

Question 6.D AIM:

Find the intersection of two sorted arrays.

OR in other words,

Given 2 sorted arrays, find all the elements which occur in both the arrays.

Input Format

- The first line contains T, the number of test cases. Following T lines contain:

1. Line 1 contains N1, followed by N1 integers of the first array
2. Line 2 contains N2, followed by N2 integers of the second array

Output Format

The intersection of the arrays in a single line

Example

Input:

1

3 10 17 57

6 2 7 10 15 57 246

Output:

10 57

Input:

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

ALGORITM:

Step 1: Start

Step 2: Input the number of test cases t.

Step 3: For each test case, input the size n1 of the first array and input the array arr1.

Step 4: Input the size n2 of the second array and input the array arr2.

Step 5: Initialize two indices i and j to 0 and use them to traverse both arrays.

Step 6: If arr1[i] < arr2[j], increment i. If arr2[j] < arr1[i], increment j.

Step 7: If arr1[i] == arr2[j], print the common element and increment both i and j.

Step 8: Continue until one of the arrays is completely traversed.

Step 9: Stop

PROGRAM:

```c
#include <stdio.h>
void intersection(int arr1[], int n1, int arr2[], int n2) {
    int i=0,j=0;
    while (i<n1 && j<n2){
        if (arr1[i]<arr2[j]){
            i++;
        }
        else if (arr2[j]<arr1[i]){
            j++;
        }
        else{
            printf("%d ",arr1[i]);
            i++;
            j++;
        }
    }
    printf("\n");
}

int main(){
    int t;
    scanf("%d",&t);
    while (t--){
        int n1,n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i=0;i<n1;i++){
            scanf("%d",&arr1[i]);
        }
        scanf("%d",&n2);
        int arr2[n2];
        for (int i=0;i<n2;i++){
            scanf("%d", &arr2[i]);
        }
        intersection(arr1,n1,arr2,n2);
    }
}
```

OUTPUT:

	Input	Expected	Got	
✔	1 3 10 17 57 6 2 7 10 15 57 246	10 57	10 57	✔
✔	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

Question 6.E AIM:

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as 5 - 1 = 4

So Return 1.

ALGORITHM:

Step 1: Start

Step 2: Input the integer n (number of elements) and the array arr.

Step 3: Input the integer k (difference to check for).

Step 4: Use a nested loop to compare each pair of elements arr[i] and arr[j].

Step 5: If the difference arr[j] - arr[i] == k, return 1.

Step 6: If the difference exceeds k, break the inner loop.

Step 7: If no valid pair is found, return 0.

Step 8: Output the result.

Step 9: Stop

PROGRAM :

```c
#include <stdio.h>
int checkpair(int arr[],int n,int k){
    for (int i=0;i<n;i++){
        for (int j=i+1;j<n;j++){
            if(arr[j]-arr[i]==k){
                return 1;
            }
            else if(arr[j]-arr[i]>k){
                break;
            }
        }
    }
    return 0;
}

int main(){
    int n, k;
    scanf("%d", &n);
    int arr[n];
    for (int i=0;i<n;i++) {
        scanf("%d",&arr[i]);
    }
    scanf("%d",&k);
    int result=checkpair(arr,n,k);
    printf("%d\n",result);
}
```

OUTPUT :

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

RESULT:

The above program is executed successfully.

Question 6.F

AIM:

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[j] - A[i] = k, i != j.

Input Format:

First Line n - Number of elements in an array

Next n Lines - N elements in the array

k - Non - Negative Integer

Output Format:

1 - If pair exists

0 - If no pair exists

Explanation for the given Sample Testcase:

YES as 5 - 1 = 4

So Return 1.

ALGORITHM:

Step 1: Start

Step 2: Input the integer n (number of elements) and the array arr.

Step 3: Input the integer k (difference to check for).

Step 4: Initialize two indices i = 0 and j = 1.

Step 5: While j < n, calculate the difference arr[j] - arr[i].

Step 6: If the difference is k, return 1.

Step 7: If the difference is less than k, increment j. If the difference is greater, increment i.

Step 8: If i == j, increment j to avoid comparing the same element with itself.

Step 9: If no valid pair is found, return 0.

Step 10: Output the result.

Step 11: Stop

PROGRAM:

```c
#include <stdio.h>
int checkpair(int arr[],int n,int k){
    int i=0,j=1;
    while(j<n){
        int diff=arr[j]-arr[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,k;
    scanf("%d",&n);
    int arr[n];
    for (int i=0;i<n;i++){
        scanf("%d",&arr[i]);
    }
    scanf("%d",&k);
    int result=checkpair(arr,n,k);
    printf("%d\n",result);
}
```

OUTPUT :

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔
✔	10 1 4 6 8 12 14 15 20 21 25 1	1	1	✔
✔	10 1 2 3 5 11 14 16 24 28 29 0	0	0	✔
✔	10 0 2 3 7 13 14 15 20 24 25 10	1	1	✔

Passed all tests! ✔

RESULT :

The above program is executed successfully.