

# COMPETITIVE PROGRAMMING:-

1) 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<br>1 1 2 3 4 | 1      |

```
1 #include <stdio.h>
2 int findDuplicate(int arr[], int n) {
3     int count[n + 1];
4     for (int i = 0; i <= n; ++i)
5         count[i] = 0;
6     for (int i = 0; i < n; ++i) {
7         if (count[arr[i]] == 1)
8             return arr[i];
9         count[arr[i]]++;
10    }
11    return -1;
12 }
13 int main() {
14     int n;
15     scanf("%d", &n);
16     int arr[n];
17     for (int i = 0; i < n; ++i)
18         scanf("%d", &arr[i]);
19     int repeatedNumber = findDuplicate(arr, n);
20     printf("%d\n", repeatedNumber);
21     return 0;
22 }
23
```

|   | Input                        | Expected | Got |   |
|---|------------------------------|----------|-----|---|
| ✓ | 11<br>10 9 7 6 5 1 2 3 8 4 7 | 7        | 7   | ✓ |
| ✓ | 5<br>1 2 3 4 4               | 4        | 4   | ✓ |
| ✓ | 5<br>1 1 2 3 4               | 1        | 1   | ✓ |

Passed all testcases. ✓

## 2) 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<br>1 1 2 3 4 | 1      |

```
1 #include <stdio.h>
2 int findDuplicate(int arr[], int n) {
3     int count[n + 1];
4     for (int i = 0; i <= n; ++i)
5         count[i] = 0;
6     for (int i = 0; i < n; ++i) {
7         if (count[arr[i]] == 1)
8             return arr[i];
9         count[arr[i]]++;
10    }
11    return -1;
12 }
13 int main() {
14     int n;
15     scanf("%d", &n);
16     int arr[n];
17     for (int i = 0; i < n; ++i)
18         scanf("%d", &arr[i]);
19
20     int repeatedNumber = findDuplicate(arr, n);
21     printf("%d\n", repeatedNumber);
22     return 0;
23 }
24
```

|   | Input                        | Expected | Got |   |
|---|------------------------------|----------|-----|---|
| ✓ | 11<br>10 9 7 6 5 1 2 3 8 4 7 | 7        | 7   | ✓ |
| ✓ | 5<br>1 2 3 4 4               | 4        | 4   | ✓ |
| ✓ | 5<br>1 1 2 3 4               | 1        | 1   | ✓ |

3) 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

**For example:**

| Input                                    | Result |
|--|--------|
| 1<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57  |

| Input | Result |
|-------|--------|
|       |        |

```
1 #include <stdio.h>
2 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
3     int i = 0, j = 0;
4     int first = 1;
5     while (i < n1 && j < n2) {
6         if (arr1[i] < arr2[j]) {
7             i++;
8         } else if (arr1[i] > arr2[j]) {
9             j++;
10        } else {
11            if (first) {
12                printf("%d", arr1[i]);
13                first = 0;
14            } else {
15                printf(" %d", arr1[i]);
16            }
17            i++;
18            j++;
19        }
20    }
21    if (first) {
22        printf(" ");
23    }
24 }
```

```

25  int main() {
26      int T;
27      scanf("%d", &T);
28      while (T--) {
29          int n1;
30          scanf("%d", &n1);
31          int arr1[n1];
32          for (int i = 0; i < n1; i++) {
33              scanf("%d", &arr1[i]);
34          }
35          int n2;
36          scanf("%d", &n2);
37          int arr2[n2];
38
39          for (int i = 0; i < n2; i++) {
40              scanf("%d", &arr2[i]);
41          }
42          findIntersection(arr1, n1, arr2, n2);
43          printf("\n");
44      }
45      return 0;
46  }

```

|   | Input                                    | Expected | Got   |   |
|---|--|----------|-------|---|
| ✓ | 1<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57    | 10 57 | ✓ |
| ✓ | 1<br>6 1 2 3 4 5 6<br>2<br>1 6           | 1 6      | 1 6   | ✓ |

4) 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

For example:

| Input            | Result |
|------------------|--------|
| 1                | 10 57  |
| 3 10 17 57       |        |
| 6                |        |
| 2 7 10 15 57 246 |        |

```
1 #include <stdio.h>
2 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
3     int i = 0, j = 0;
4     int first = 1;
5     while (i < n1 && j < n2) {
6         if (arr1[i] < arr2[j]) {
7             i++;
8         } else if (arr1[i] > arr2[j]) {
9             j++;
10        } else {
11            if (first) {
12                printf("%d", arr1[i]);
13                first = 0;
14            } else {
15                printf(" %d", arr1[i]);
16                i++;
17                j++;
18            }
19        }
20    }
```

```

21 ▾ int main() {
22     int T;
23     scanf("%d", &T);
24
25 ▾     while (T--) {
26         int n1;
27         scanf("%d", &n1);
28         int arr1[n1];
29
30 ▾         for (int i = 0; i < n1; i++) {
31             scanf("%d", &arr1[i]);
32         }
33         int n2;
34         scanf("%d", &n2);
35         int arr2[n2];
36
37 ▾         for (int i = 0; i < n2; i++) {
38             scanf("%d", &arr2[i]);
39         }
40         findIntersection(arr1, n1, arr2, n2);
41         printf("\n");
42     }
43     return 0;
44 }

```

|   | Input                                    | Expected | Got   |   |
|---|--|----------|-------|---|
| ✓ | 1<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57    | 10 57 | ✓ |
| ✓ | 1<br>6 1 2 3 4 5 6<br>2                  | 1 6      | 1 6   | ✓ |

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

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.

For example:

| Input | Result |
|-------|--------|
| 3     | 1      |
| 1 3 5 |        |
| 4     |        |

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 void array(int n,int a[],int k){
4     for(int i=0;i<n;i++){
5         for(int j=i+1;j<n;j++){
6             if(abs(a[i]-a[j])==k){
7                 printf("1");
8                 return;
9             }
10        }
11    }
12    printf("0");
13 }
14 int main(){
15     int n,k;
16     scanf("%d",&n);
17     int a[n];
18     for(int i=0;i<n;i++){
19         scanf("%d",&a[i]);
20     }
21     scanf("%d",&k);
22     array(n,a,k);
23     return 0;
24 }
```



|   | Input                                 | Expected | Got |   |
|---|---------------------------------------|----------|-----|---|
| ✓ | 3<br>1 3 5<br>4                       | 1        | 1   | ✓ |
| ✓ | 10<br>1 4 6 8 12 14 15 20 21 25<br>1  | 1        | 1   | ✓ |
| ✓ | 10<br>1 2 3 5 11 14 16 24 28 29<br>0  | 0        | 0   | ✓ |
| ✓ | 10<br>0 2 3 7 13 14 15 20 24 25<br>10 | 1        | 1   | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

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

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.

For example:

| Input | Result |
|-------|--------|
| 3     | 1      |
| 1 3 5 |        |
| 4     |        |

```
1 #include <stdio.h>
2 int main() {
3     int n, k;
4     scanf("%d", &n);
5     int a[n];
6     for (int i = 0; i < n; i++) {
7         scanf("%d", &a[i]);
8     }
9     scanf("%d", &k);
10    for (int i = 0; i < n; i++) {
11        for (int j = i + 1; j < n; j++) {
12            if (a[j] - a[i] == k) {
13                printf("1\n");
14                return 0;
15            }
16        }
17    }
18    printf("0\n");
19    return 0;
20 }
```

|   | Input                                 | Expected | Got |   |
|---|---------------------------------------|----------|-----|---|
| ✓ | 3<br>1 3 5<br>4                       | 1        | 1   | ✓ |
| ✓ | 10<br>1 4 6 8 12 14 15 20 21 25<br>1  | 1        | 1   | ✓ |
| ✓ | 10<br>1 2 3 5 11 14 16 24 28 29<br>0  | 0        | 0   | ✓ |
| ✓ | 10<br>0 2 3 7 13 14 15 20 24 25<br>10 | 1        | 1   | ✓ |

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.