



Dashboard My courses



CS23331-DAA-2024-CSE / 1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Complexity

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

Started on	Monday, 3 November 2025, 6:07 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:08 AM
Time taken	1 min 7 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 

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

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int findDuplicate(int arr[], int n) {
4     int slow = arr[0];
5     int fast = arr[0];
6
7     // Phase 1: Detect cycle
8     do {
9         slow = arr[slow];
10        fast = arr[arr[fast]];
11    } while (slow != fast);
12
13    // Phase 2: Find entry point of cycle (duplicate)
14    slow = arr[0];
15    while (slow != fast) {
16        slow = arr[slow];
17        fast = arr[fast];
18    }
19
20    return slow;
21}
22
23 int main() {
24     int n;
25     scanf("%d", &n);
26
27     int arr[n];
28     for (int i = 0; i < n; i++)
29         scanf("%d", &arr[i]);
30
31     printf("%d\n", findDuplicate(arr, n));
32
33     return 0;
34}
35
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[Back to Course](#)

Data retention summary



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

Started on	Monday, 3 November 2025, 6:09 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:10 AM
Time taken	1 min 24 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 

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 1 2 3 4	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++) {
9         scanf("%d", &arr[i]);
10    }
11
12    int sum = 0;
13    for (int i = 0; i < n; i++)
14        sum += arr[i];
15
16    // Sum of numbers from 1 to n-1 (since one repeats)
17    int expected_sum = (n - 1) * n / 2;
18
19    int duplicate = sum - expected_sum;
20
21    printf("%d\n", duplicate);
22
23    return 0;
24}
25
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

[Back to Course](#)

Data retention summary



Dashboard My courses



CS23331-DAA-2024-CSE / 3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Space Complexity

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

Started on	Monday, 3 November 2025, 6:10 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:12 AM
Time taken	1 min 38 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 

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:
 - Line 1 contains N1, followed by N1 integers of the first array
 - 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	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T);
6
7     while (T--) {
8         int n1, n2;
9         scanf("%d", &n1);
10        int a[n1];
11        for (int i = 0; i < n1; i++)
12            scanf("%d", &a[i]);
13
14        scanf("%d", &n2);
15        int b[n2];
16        for (int i = 0; i < n2; i++)
17            scanf("%d", &b[i]);
18
19        int i = 0, j = 0;
20        int found = 0;
21
22        // Two-pointer traversal
23        while (i < n1 && j < n2) {
```

```
24
25     if (a[i] == b[j]) {
26         printf("%d ", a[i]);
27         found = 1;
28         i++;
29         j++;
30     } else if (a[i] < b[j]) {
31         i++;
32     } else {
33         j++;
34     }
35
36     if (!found)
37         printf("No Intersection");
38
39     printf("\n");
40 }
41
42 return 0;
43 }
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[Back to Course](#)

Data retention summary



 Dashboard My courses 

CS23331-DAA-2024-CSE / 4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) Space Complexity

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

Started on	Monday, 3 November 2025, 6:12 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:14 AM
Time taken	1 min 55 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 

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:
 - Line 1 contains N1, followed by N1 integers of the first array
 - 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	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T); // number of test cases
6
7     while (T--) {
8         int n1, n2;
9
10        // Read first array
11        scanf("%d", &n1);
12        int arr1[n1];
13        for (int i = 0; i < n1; i++)
14            scanf("%d", &arr1[i]);
15
16        // Read second array
17        scanf("%d", &n2);
18        int arr2[n2];
19        for (int i = 0; i < n2; i++)
20            scanf("%d", &arr2[i]);
21
22        int i = 0, j = 0;
23        int found = 0;
```

```

24
25    // Two-pointer approach
26    while (i < n1 && j < n2) {
27        if (arr1[i] == arr2[j]) {
28            printf("%d ", arr1[i]);
29            found = 1;
30            i++;
31            j++;
32        } else if (arr1[i] < arr2[j]) {
33            i++;
34        } else {
35            j++;
36        }
37    }
38
39    if (!found)
40        printf("No Intersection");
41
42    printf("\n");
43}
44
45    return 0;
46}
47

```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[Back to Course](#)

Data retention summary



Dashboard My courses



CS23331-DAA-2024-CSE / 5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Complexity

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

Started on	Monday, 3 November 2025, 6:14 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:15 AM
Time taken	1 min 17 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 

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	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++)
9         scanf("%d", &arr[i]);
10
11    int k;
12    scanf("%d", &k);
13
14    int i = 0, j = 1;
15    int found = 0;
16
17    while (i < n && j < n) {
18        if (i != j) {
19            int diff = arr[j] - arr[i];
20
21            if (diff == k) {
22                found = 1;
23                break;
24            } else if (diff < k) {
25                j++;
26            } else {
27                i++;
28            }
29        } else {
30            j++;
31        }
32    }
33
34    printf("%d\n", found);
35
36    return 0;
37 }
```

	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! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[Back to Course](#)

Data retention summary



Dashboard My courses



CS23331-DAA-2024-CSE / 6-Pair with Difference -O(n) Time Complexity,O(1) Space Complexity

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

Started on	Monday, 3 November 2025, 6:16 AM
State	Finished
Completed on	Monday, 3 November 2025, 6:17 AM
Time taken	1 min 9 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 

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	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int n;
5     scanf("%d", &n);
6
7     int arr[n];
8     for (int i = 0; i < n; i++)
9         scanf("%d", &arr[i]);
10
11    int k;
12    scanf("%d", &k);
13
14    int i = 0, j = 1;
15    int found = 0;
16
17    // Two-pointer approach
18    while (i < n && j < n) {
19        int diff = arr[j] - arr[i];
20
21        if (i != j && diff == k) {
22            found = 1;
23            break;
24        } else if (diff < k) {
25            j++;
26        } else {
27            i++;
28        }
29    }
30
31    printf("%d\n", found);
32
33    return 0;
34 }
```

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! ✓

Correct

Marks for this submission: 1.00/1.00.

[Finish review](#)

[Back to Course](#)

Data retention summary