



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

Started on	Friday, 10 October 2025, 2:19 PM
State	Finished
Completed on	Friday, 10 October 2025, 2:22 PM
Time taken	3 mins 38 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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 findDuplicate(int nums[], int n) {
4     int tortoise = nums[0];
5     int hare = nums[0];
6
7     do {
8         tortoise = nums[tortoise];
9         hare = nums[nums[hare]];
10    } while (tortoise != hare);
11
12    tortoise = nums[0];
13    while (tortoise != hare) {
14        tortoise = nums[tortoise];
15        hare = nums[hare];
16    }
17
18    return hare;
19}
20
21 int main() {
22     int n;
23     scanf("%d", &n);
24
25     int nums[n];
26     for (int i = 0; i < n; i++) {
27         scanf("%d", &nums[i]);
28     }
29
30     printf("%d\n", findDuplicate(nums, n));
31     return 0;
32 }
```

	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



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



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

Started on	Friday, 10 October 2025, 2:24 PM
State	Finished
Completed on	Friday, 10 October 2025, 2:28 PM
Time taken	4 mins 5 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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 findDuplicate(int nums[], int n) {
4     int tortoise = nums[0];
5     int hare = nums[0];
6
7     do {
8         tortoise = nums[tortoise];
9         hare = nums[nums[hare]];
10    } while (tortoise != hare);
11
12    tortoise = nums[0];
13    while (tortoise != hare) {
14        tortoise = nums[tortoise];
15        hare = nums[hare];
16    }
17
18    return hare;
19}
20
21 int main() {
22     int n;
23     scanf("%d", &n);
24
25     int nums[n];
26     for (int i = 0; i < n; i++) {
27         scanf("%d", &nums[i]);
28     }
29
30     printf("%d\n", findDuplicate(nums, n));
31     return 0;
32 }
```

	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](#)



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	Friday, 10 October 2025, 2:29 PM
State	Finished
Completed on	Friday, 10 October 2025, 2:32 PM
Time taken	2 mins 48 secs
Marks	1.00/1.00
Grade	<b>30.00</b> out of 30.00 ( <b>100%</b> )

**Question 1** | Correct Mark 1.00 out of 1.00 Flag question

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     while (T--) {
7         int N1, N2;
8         scanf("%d", &N1);
9         int arr1[N1];
10    for (int i = 0; i < N1; i++) {
11        scanf("%d", &arr1[i]);
12    }
13
14    scanf("%d", &N2);
15    int arr2[N2];
16    for (int i = 0; i < N2; i++) {
17        scanf("%d", &arr2[i]);
18    }
19
20    int i = 0, j = 0;
21    int last_printed = -1; // to avoid duplicates; assumes inputs > 0
22    int first = 1; // to handle spacing in output
23
24    while (i < N1 && j < N2) {
25        if (arr1[i] == arr2[j]) {
26            if (arr1[i] != last_printed) {
27                if (!first) printf(" ");
28                printf("%d", arr1[i]);
29                last_printed = arr1[i];
30                first = 0;
31            }
32            i++; j++;
33        } else if (arr1[i] < arr2[j]) {
34            i++;
35        } else {
36            j++;
37        }
38    }
39    printf("\n");
40}
41    return 0;
42}
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



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	Friday, 10 October 2025, 2:33 PM
State	Finished
Completed on	Friday, 10 October 2025, 2:40 PM
Time taken	6 mins 41 secs
Marks	1.00/1.00
Grade	<b>30.00</b> out of 30.00 ( <b>100%</b> )

**Question 1** | Correct Mark 1.00 out of 1.00 Flag question

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     while (T--) {
7         int N1, N2;
8         scanf("%d", &N1);
9         int arr1[N1];
10    for (int i = 0; i < N1; i++) {
11        scanf("%d", &arr1[i]);
12    }
13
14    scanf("%d", &N2);
15    int arr2[N2];
16    for (int i = 0; i < N2; i++) {
17        scanf("%d", &arr2[i]);
18    }
19
20    int i = 0, j = 0;
21    int last_printed = -1; // assuming input elements >= 0
22    int first_printed = 0;
23
24    while (i < N1 && j < N2) {
25        if (arr1[i] == arr2[j]) {
26            if (arr1[i] != last_printed) {
27                if (first_printed) printf(" ");
28                printf("%d", arr1[i]);
29                last_printed = arr1[i];
30                first_printed = 1;
31            }
32            i++;
33            j++;
34        } else if (arr1[i] < arr2[j]) {
35            i++;
36        } else {
37            j++;
38        }
39    }
40    printf("\n");
41 }
42 return 0;
43 }
44 }
```

	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



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	Friday, 10 October 2025, 5:24 PM
State	Finished
Completed on	Friday, 10 October 2025, 5:30 PM
Time taken	5 mins 39 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100%)

Question 1 | Correct Mark 1.00 out of 1.00 Flag question

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, k;
5     scanf("%d", &n);
6     int A[n];
7     for (int i = 0; i < n; i++) {
8         scanf("%d", &A[i]);
9     }
10    scanf("%d", &k);
11    int i = 0, j = 1, found = 0;
12    while (i < n && j < n) {
13        if (i != j) {
14            int diff = A[j] - A[i];
15            if (diff == k) {
16                found = 1;
17                break;
18            } else if (diff < k) {
19                j++;
20            } else {
21                i++;
22            }
23        } else {
```

```
24     j++;
25 }
26 }
27 printf("%d\n", found);
28 return 0;
29 }
30 }
```

	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



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	Friday, 10 October 2025, 5:31 PM
State	Finished
Completed on	Friday, 10 October 2025, 5:35 PM
Time taken	4 mins 5 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, k;
5     scanf("%d", &n);
6     int A[n];
7     for (int i = 0; i < n; i++) {
8         scanf("%d", &A[i]);
9     }
10    scanf("%d", &k);
11    int i = 0, j = 1, found = 0;
12    while (i < n && j < n) {
13        if (i != j) {
14            int diff = A[j] - A[i];
15            if (diff == k) {
16                found = 1;
17                break;
18            } else if (diff < k) {
19                j++;
20            } else {
21                i++;
22            }
23        } else {
```

```
24     j++;
25 }
26 }
27 printf("%d\n", found);
28 return 0;
29 }
30 }
```

	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