



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

Started on Monday, 13 October 2025, 2:10 PM

State Finished

Completed on Monday, 13 October 2025, 2:24 PM

Time taken 14 mins 33 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     if (n <= 1) {
5         return -1;
6     }
7     int tortoise = nums[0];
8     int hare = nums[nums[0]];
9     while (tortoise != hare) {
10        tortoise = nums[tortoise];
11        hare = nums[nums[hare]];
12    }
13    int ptr1 = 0;
14    int ptr2 = tortoise;
15    while (ptr1 != ptr2) {
16        ptr1 = nums[ptr1];
17        ptr2 = nums[ptr2];
18    }
19    return ptr1;
20}
21
22 int main() {
23     int n;
24     scanf("%d", &n);
25     int nums[n];
26     for (int i = 0; i < n; i++) {
27         scanf("%d", &nums[i]);
28     }
29     printf("%d\n", findDuplicate(nums, n));
30     return 0;
31 }
```

	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](#)



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

Started on	Monday, 13 October 2025, 2:25 PM
State	Finished
Completed on	Monday, 13 October 2025, 2:35 PM
Time taken	10 mins 48 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     int findDuplicate(int* nums, int n) {
3         if (n <= 1) {
4             return -1;
5         }
6         int tortoise = nums[0];
7         int hare = nums[nums[0]];
8         while (tortoise != hare) {
9             tortoise = nums[tortoise];
10            hare = nums[nums[hare]];
11        }
12        int ptr1 = 0;
13        int ptr2 = tortoise;
14        while (ptr1 != ptr2) {
15            ptr1 = nums[ptr1];
16            ptr2 = nums[ptr2];
17        }
18        return ptr1;
19    }
20
21 int main() {
22     int n;
23     scanf("%d", &n);
24     int nums[n];
25     for (int i = 0; i < n; i++) {
26         scanf("%d", &nums[i]);
27     }
28     printf("%d\n", findDuplicate(nums, n));
29     return 0;
30 }
```

	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](#)

CS23331-DAA-2024-AIML / 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, 13 October 2025, 2:36 PM
State	Finished
Completed on	Monday, 13 October 2025, 2:45 PM
Time taken	9 mins 24 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

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
10 57
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 #include <stdlib.h>
3
4 void findIntersection(int* arr1, int n1, int* arr2, int n2) {
5     int i = 0;
6     int j = 0;
7     int first_element = 1;
8
9     while (i < n1 && j < n2) {
10        if (arr1[i] < arr2[j]) {
11            i++;
12        } else if (arr2[j] < arr1[i]) {
13            j++;
14        } else {
15            if (first_element == 0) {
16                printf(" ");
17            }
18            printf("%d", arr1[i]);
19            first_element = 0;
20            i++;
21            j++;
22        }
23    }
24    printf("\n");
25 }
26
27 void solve() {
28     int n1, n2;
29
30     if (scanf("%d", &n1) != 1) return;
31     int* arr1 = (int*)malloc(n1 * sizeof(int));
32     for (int k = 0; k < n1; k++) {
33         if (scanf("%d", &arr1[k]) != 1) {
34             free(arr1);
35             return;
36         }
37     }
38
39     if (scanf("%d", &n2) != 1) {
40         free(arr1);
41         return;
42     }
43     int* arr2 = (int*)malloc(n2 * sizeof(int));
44     for (int k = 0; k < n2; k++) {
45         if (scanf("%d", &arr2[k]) != 1) {
46             free(arr1);
47             free(arr2);
48             return;
49         }
50     }
51     findIntersection(arr1, n1, arr2, n2);
52 }
```

	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 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course

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

Started on	Monday, 27 October 2025, 1:21 PM
State	Finished
Completed on	Monday, 27 October 2025, 1:29 PM
Time taken	8 mins 35 secs
Marks	1.00/1.00
Grade	30.00 out of 30.00 (100%)

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 3 10 17 57 6 2 7 10 15 57 246	10 57

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        int n2;
13        scanf("%d", &n2);
14        int arr2[n2];
15        for (int i = 0; i < n2; i++)
16            scanf("%d", &arr2[i]);
17        int i = 0, j = 0, first = 1;
18        while (i < n1 && j < n2) {
19            if (arr1[i] == arr2[j]) {
20                if (!first) printf(" ");
21                printf("%d", arr1[i]);
22                first = 0;
23                i++;
24                j++;
25            } else if (arr1[i] < arr2[j]) {
26                i++;
27            } else {
28                j++;
29            }
30            printf("\n");
31        }
32    return 0;
33 }
```

	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 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Finish review

Back to Course



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

Started on Monday, 27 October 2025, 1:22 PM

State Finished

Completed on Monday, 27 October 2025, 1:33 PM

Time taken 10 mins 53 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 != 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     int arr[n];
7     for (int i = 0; i < n; i++)
8         scanf("%d", &arr[i]);
9     int k;
10    scanf("%d", &k);
11    int i = 0, j = 1, found = 0;
12    while (i < n && j < n) {
13        int diff = arr[j] - arr[i];
14        if (diff == k && i != j) {
15            found = 1;
16            break;
17        } else if (diff < k) {
18            j++;
19        } else {
20            i++;
21        }
22    }
23    printf("%d\n", found);
24    return 0;
25 }
```

	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](#)



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

Started on	Monday, 27 October 2025, 1:46 PM
State	Finished
Completed on	Monday, 27 October 2025, 2:11 PM
Time taken	24 mins 53 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;
5     scanf("%d", &n);
6     int arr[n];
7     for (int i = 0; i < n; i++)
8         scanf("%d", &arr[i]);
9     int k;
10    scanf("%d", &k);
11    int i = 0, j = 1, found = 0;
12    while (i < n && j < n) {
13        int diff = arr[j] - arr[i];
14        if (diff == k && i != j) {
15            found = 1;
16            break;
17        } else if (diff < k) {
18            j++;
19        } else {
20            i++;
21        }
22    }
23    printf("%d\n", found);
24    return 0;
25 }
```

Passed all tests!

	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](#)