

[Dashbo...](#) / [My cour...](#) / [CS23331-DAA-2023-...](#) / [Competitive Program...](#) / [1-Finding Duplicates- \$O\(n^2\)\$ Time Complexity, \$O\(1\)\$ Space Co...](#)

Started on	Saturday, 2 November 2024, 6:10 PM
State	Finished
Completed on	Saturday, 2 November 2024, 6:13 PM
Time taken	2 mins 11 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 find_duplicate(int nums[], int n) {
4      int seen[n + 1];
5      for (int i = 0; i <= n; i++) {
6          seen[i] = 0;
7      }
8
9      for (int i = 0; i < n; i++) {
10         if (seen[nums[i]]) {
11             return nums[i];
12         }
13         seen[nums[i]] = 1;
14     }
15     return -1;
16 }
17
18 int main() {
19     int n;
20     scanf("%d", &n);
21     int nums[n];
22
23     for (int i = 0; i < n; i++) {
24         scanf("%d", &nums[i]);
25     }
26
27     int duplicate = find_duplicate(nums, n);
28     if (duplicate != -1) {
29         printf("%d\n", duplicate);
30     }
31
32     return 0;
33 }
34

```

	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.

◀ 4-DP-Longest non-decreasing Subsequence

Jump to...

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

[Dashbo...](#) / [My cour...](#) / [CS23331-DAA-2023-A...](#) / [Competitive Programm...](#) / [2-Finding Duplicates-O\(n\) Time Complexity,O\(1\) Space Comp...](#)

Started on	Saturday, 2 November 2024, 6:13 PM
State	Finished
Completed on	Saturday, 2 November 2024, 6:14 PM
Time taken	1 min 1 sec
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 find_duplicate(int nums[], int n) {
4      int seen[n + 1];
5      for (int i = 0; i <= n; i++) {
6          seen[i] = 0;
7      }
8
9      for (int i = 0; i < n; i++) {
10         if (seen[nums[i]]) {
11             return nums[i];
12         }
13         seen[nums[i]] = 1;
14     }
15     return -1;
16 }
17
18 int main() {
19     int n;
20     scanf("%d", &n);
21     int nums[n];
22
23     for (int i = 0; i < n; i++) {
24         scanf("%d", &nums[i]);
25     }
26
27     int duplicate = find_duplicate(nums, n);
28     if (duplicate != -1) {
29         printf("%d\n", duplicate);
30     }
31
32     return 0;
33 }
34

```

	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.

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

Jump to...

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

[Dashb...](#) / [My cou...](#) / [CS23331-DAA-202...](#) / [Competitive Progra...](#) / [3-Print Intersection of 2 sorted arrays- \$O\(m*n\)\$ Time Complexity, \$O\(1\)\$ S...](#)

Started on	Saturday, 2 November 2024, 6:14 PM
State	Finished
Completed on	Saturday, 2 November 2024, 6:15 PM
Time taken	42 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:

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

Answer: (penalty regime: 0 %)

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



```
16     } else {
17         printf(" %d", arr1[i]);
18     }
19     i++;
20     j++;
21 }
22 }
23 printf("\n");
24 }
25
26 int main() {
27     int T;
28     scanf("%d", &T);
29
30     while (T--) {
31         int n1;
32         scanf("%d", &n1);
33         int arr1[n1];
34
35         for (int i = 0; i < n1; i++) {
36             scanf("%d", &arr1[i]);
37         }
38
39         int n2;
40         scanf("%d", &n2);
41         int arr2[n2];
42
43         for (int i = 0; i < n2; i++) {
44             scanf("%d", &arr2[i]);
45         }
46
47         find_intersection(arr1, n1, arr2, n2);
48     }
49
50     return 0;
51 }
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 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

Jump to...

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

[Dashb...](#) / [My cou...](#) / [CS23331-DAA-202...](#) / [Competitive Progra...](#) / [4-Print Intersection of 2 sorted arrays- \$O\(m+n\)\$ Time Complexity, \$O\(1\)\$ S...](#)

Started on	Saturday, 2 November 2024, 6:15 PM
State	Finished
Completed on	Saturday, 2 November 2024, 6:17 PM
Time taken	1 min 56 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:

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

Answer: (penalty regime: 0 %)

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

```

16     } else {
17         printf(" %d", arr1[i]);
18     }
19     i++;
20     j++;
21 }
22 }
23 printf("\n");
24 }
25
26 int main() {
27     int T;
28     scanf("%d", &T);
29
30     while (T--) {
31         int n1;
32         scanf("%d", &n1);
33         int arr1[n1];
34
35         for (int i = 0; i < n1; i++) {
36             scanf("%d", &arr1[i]);
37         }
38
39         int n2;
40         scanf("%d", &n2);
41         int arr2[n2];
42
43         for (int i = 0; i < n2; i++) {
44             scanf("%d", &arr2[i]);
45         }
46
47         find_intersection(arr1, n1, arr2, n2);
48     }
49
50     return 0;
51 }
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 6 2 1 6	1 6	1 6	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

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

Jump to...

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

[Dashbo...](#) / [My cour...](#) / [CS23331-DAA-2023-...](#) / [Competitive Program...](#) / [5-Pair with Difference- \$O\(n^2\)\$ Time Complexity, \$O\(1\)\$ Space Co...](#)

Started on	Saturday, 2 November 2024, 6:18 PM
State	Finished
Completed on	Saturday, 2 November 2024, 6:18 PM
Time taken	28 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 3 5 4	1

Answer: (penalty regime: 0 %)

```

1 #include <stdio.h>
2
3 int find_pair_with_difference(int arr[], int n, int k) {
4     int i = 0, j = 0;
5
6     while (i < n && j < n) {
7         int diff = arr[j] - arr[i];
8
9         if (diff == k && i != j) {
10             return 1; // Pair found
11         } else if (diff < k) {
12             j++; // Increase j to increase the difference
13         } else {
14             i++; // Increase i to decrease the difference
15         }
16         if (i == j) {
17             j++; // Ensure j is always ahead of i
18         }
19     }
20     return 0; // No pair found
21 }
22
23 int main() {
24     int n;
25     scanf("%d", &n);
26     int arr[n];
27
28     for (int i = 0; i < n; i++) {
29         scanf("%d", &arr[i]);
30     }
31
32     int k;
33     scanf("%d", &k);

```

```
34
35     int result = find_pair_with_difference(arr, n, k);
36     printf("%d\n", result);
37
38     return 0;
39 }
40
```

	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.

[◀ 4-Print Intersection of 2 sorted arrays- \$O\(m+n\)\$ Time Complexity, \$O\(1\)\$ Space Complexity](#)

Jump to...



[6-Pair with Difference - \$O\(n\)\$ Time Complexity, \$O\(1\)\$ Space Complexity ▶](#)

[Dashbo...](#) / [My cour...](#) / [CS23331-DAA-2023-A...](#) / [Competitive Program...](#) / [6-Pair with Difference -O\(n\) Time Complexity,O\(1\) Space Com...](#)

Started on	Saturday, 2 November 2024, 6:18 PM
State	Finished
Completed on	Saturday, 2 November 2024, 6:19 PM
Time taken	35 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 3 5 4	1

Answer: (penalty regime: 0 %)

```

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

```

```
34
35     int result = find_pair_with_difference(arr, n, k);
36     printf("%d\n", result);
37
38     return 0;
39 }
40
```

	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.

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

Jump to...

