

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

Started on Tuesday, 21 October 2025, 3:53 PM

State Finished

Completed on Tuesday, 21 October 2025, 3:53 PM

Time taken 40 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 #define MAX 100001
4
5 int main() {
6     int n;
7     scanf("%d", &n);
8     int freq[MAX] = {0};
```

1 1 2 3 4

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 #define MAX 100001
4
5 int main() {
6     int n;
7     scanf("%d", &n);
8
9     int freq[MAX] = {0};
10    int x;
11
12    for (int i = 0; i < n; i++) {
13        scanf("%d", &x);
14        if (freq[x] == 1) {
15            printf("%d\n", x);
16            return 0;
17        }
18        freq[x]++;
19    }
20
21    return 0;
22}
23
```

	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](#)

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

Started on	Tuesday, 21 October 2025, 3:54 PM
State	Finished
Completed on	Tuesday, 21 October 2025, 3:55 PM
Time taken	29 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 #define MAX 100001
4
5 int main() {
6     int n;
7     scanf("%d", &n);
8 }
```

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 #define MAX 100001
4
5 int main() {
6     int n;
7     scanf("%d", &n);
8
9     int freq[MAX] = {0};
10    int x;
11
12    for (int i = 0; i < n; i++) {
13        scanf("%d", &x);
14        if (freq[x] == 1) {
15            printf("%d\n", x);
16            return 0;
17        }
18        freq[x]++;
19    }
20
21    return 0;
22 }
23 }
```

	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](#)

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

Started on Tuesday, 21 October 2025, 3:55 PM

State Finished

Completed on Tuesday, 21 October 2025, 3:56 PM

Time taken 1 min 2 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:

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	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 #define MAX 10000
4
5 int main() {
6     int T;
7     scanf("%d", &T);
8
9     while (T--) {
10         int N1, N2;
11         int A[MAX], B[MAX];
12
13         scanf("%d", &N1);
14         for (int i = 0; i < N1; i++) {
15             scanf("%d", &A[i]);
16         }
17
18         scanf("%d", &N2);
19         for (int i = 0; i < N2; i++) {
20             scanf("%d", &B[i]);
21         }
22
23         int i = 0, j = 0, last_printed = -1000000000;
24         while (i < N1 && j < N2) {
25             if (A[i] == B[j]) {
26                 if (A[i] != last_printed) {
27                     printf("%d ", A[i]);
28                     last_printed = A[i];
29                 }
30                 i++;
31                 j++;
32             } else if (A[i] < B[j]) {
33                 i++;
34             }
35         }
36     }
37 }
```

```

16
17
18     scanf("%d", &N2);
19     for (int i = 0; i < N2; i++) {
20         scanf("%d", &B[i]);
21     }
22
23     int i = 0, j = 0, last_printed = -1000000000;
24     while (i < N1 && j < N2) {
25         if (A[i] == B[j]) {
26             if (A[i] != last_printed) {
27                 printf("%d ", A[i]);
28                 last_printed = A[i];
29             }
30             i++;
31             j++;
32         } else if (A[i] < B[j]) {
33             i++;
34         } else {
35             j++;
36         }
37     }
38     printf("\n");
39 }
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.

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

Started on Tuesday, 21 October 2025, 3:56 PM

State Finished

Completed on Tuesday, 21 October 2025, 3:57 PM

Time taken 37 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	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>  
2  
3 #define MAX 10000  
4  
5 int main() {  
6     int T;  
7     scanf("%d", &T);  
8  
9     while (T--) {  
10         int N1, N2;  
11         int A[MAX], B[MAX];  
12  
13         scanf("%d", &N1);  
14         for (int i = 0; i < N1; i++) {  
15             scanf("%d", &A[i]);  
16         }  
17  
18         scanf("%d", &N2);  
19         for (int i = 0; i < N2; i++) {  
20             scanf("%d", &B[i]);  
21         }  
22  
23         int i = 0, j = 0;  
24         int last = -1000000000;  
25  
26         while (i < N1 && j < N2) {  
27             if (A[i] == B[j]) {  
28                 if (A[i] != last) {  
29                     printf("%d ", A[i]);  
30                     last = A[i];  
31                 }  
32                 i++;  
33                 j++;  
34             } else if (A[i] < B[j]) {  
35                 i++;  
36             } else {  
37                 j++;  
38             }  
39         }  
40         printf("\n");  
41     }  
42 }
```

```

19+
20    for (cntc i = 0, i < N2, i++) {
21        scanf("%d", &B[i]);
22    }
23
24    int i = 0, j = 0;
25    int last = -1000000000;
26
27    while (i < N1 && j < N2) {
28        if (A[i] == B[j]) {
29            if (A[i] != last) {
30                printf("%d ", A[i]);
31                last = A[i];
32            }
33            i++;
34        } else if (A[i] < B[j]) {
35            i++;
36        } else {
37            j++;
38        }
39    }
40    printf("\n");
41}
42
43    return 0;
44}
45

```

	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](#)

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

Started on Tuesday, 21 October 2025, 3:57 PM

State Finished

Completed on Tuesday, 21 October 2025, 3:58 PM

Time taken 41 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	

```

1 #include <stdio.h>
2
3 #define MAX 100000
4
5 int main() {
6     int n, k;
7     int A[MAX];
8
9     scanf("%d", &n);
10    for (int i = 0; i < n; i++) {
11        scanf("%d", &A[i]);
12    }
13    scanf("%d", &k);
14
15    int i = 0, j = 1;
16
17    while (i < n && j < n) {
18        if (i != j && A[j] - A[i] == k) {
19            printf("1\n");
20            return 0;
21        } else if (A[j] - A[i] < k) {
22            j++;
23        } else {
24            i++;
25        }
26    }
27
28    printf("0\n");
29    return 0;
30 }
31

```

	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	✓

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

Started on	Tuesday, 21 October 2025, 3:58 PM
State	Finished
Completed on	Tuesday, 21 October 2025, 3:58 PM
Time taken	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](#)

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>
```

```

3 #define MAX 100000
4
5 int main() {
6     int n, k;
7     int A[MAX];
8
9     scanf("%d", &n);
10    for (int i = 0; i < n; i++) {
11        scanf("%d", &A[i]);
12    }
13    scanf("%d", &k);
14
15    int i = 0, j = 1;
16
17    while (i < n && j < n) {
18        int diff = A[j] - A[i];
19        if (i != j && diff == k) {
20            printf("1\n");
21            return 0;
22        } else if (diff < k) {
23            j++;
24        } else {
25            i++;
26        }
27    }
28
29    printf("0\n");
30    return 0;
31 }
32

```

	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

Last execution took 4.000 ms