

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

Started on: Tuesday, 21 October 2025, 5:58 PM

State: Finished

Completed on: Tuesday, 21 October 2025, 5:58 PM

Time taken: 39 sec

Marks: 1.00/1.00

Grade: 4.00 out of 4.00 (100%)

Question 1: Correct. Marks: 1.00 out of 1.00. [Flag for review](#)

Given an array A of sorted integers and another non-negative integer k, And if there exists 2 indices i and j such that  $|A[i] - A[j]| = k$ , i < j.

Input Format:

First Line n - Number of elements in an array

Next n Lines - 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  $1 - 7 = 6$ 

So Return 1.

For example:

Input: Result

5	1
1 2 3	
4	

Answer: (correctly) Integer: 0 10

```

1  // O(n) Time Complexity, O(1) Space Complexity
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

```

```

1 // O(n) Time Complexity, O(1) Space Complexity
2
3 // Input: n, k, A
4 int n, k;
5 int A[100000];
6
7 // Read input
8 scanf("%d", &n);
9 for (int i = 0; i < n; i++) {
10     scanf("%d", &A[i]);
11 }
12
13 // Read k
14 scanf("%d", &k);
15
16 // Initialize i, j
17 int i = 0, j = 1;
18
19 // While loop to find the pair
20 while (i < n && j < n) {
21     int diff = A[j] - A[i];
22     if (diff == k) {
23         printf("Yes\n");
24         return 0;
25     } else if (diff < k) {
26         i++;
27     } else {
28         j++;
29     }
30 }
31
32 // If no pair found
33 printf("No\n");
34 return 0;
35 }

```

Input:	Expected	Got
5 1 2 3 4	1	1 ✓
10 0 4 6 9 22 14 19 28 21 25 5	1	1 ✓
10 1 2 3 5 11 14 18 24 29 28 0	0	0 ✓
10 0 2 3 7 12 14 25 28 24 25 10	1	1 ✓

Passed all tests ✓

Correct

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

Started on: Tuesday, 21 October 2025, 9:57 PM

Status: Finished

Completed on: Tuesday, 21 October 2025, 9:58 PM

Time taken: 47 secs

Marks: 1.00/1.00

Grade: 4.00 out of 4.00 (100%)

Question 1: Correct. Marks: 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  $A[j] - A[i] = k$ ,  $1 \leq j$ .

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:

VS as  $5 - 1 = 4$

So Return 1.

For example:

Input	Result
3 1 3 5 4	1

```

1 // Problem: pairWithK.cpp
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 (j < n && i < n) {
18        if (A[j] - A[i] == k) {
19            printf("Yes\n");
20            return 0;
21        } else if (A[j] - A[i] < k) {
22            i++;
23        } else {
24            j++;
25        }
26    }
27
28    printf("No\n");
29    return 0;
30 }

```

Input	Expected	Got
3 1 3 5 4	1	1 ✓
10 2 4 6 8 12 14 15 18 21 25 9	1	1 ✓
10 3 3 3 3 11 14 15 24 28 29 9	0	0 ✓
10 6 2 3 7 13 14 15 18 24 25 18	1	1 ✓

```

20         cout<<"No", AB(1,2);
21     }
22     let i = 0, j = 0;
23     let last = -1000000000;
24
25     while (i < N1 && j < N2) {
26         if (A[i] == B[j]) {
27             cout<<"Yes";
28             last = A[i];
29         }
30         i++;
31         j++;
32     } else if (A[i] < B[j]) {
33         i++;
34     } else {
35         j++;
36     }
37 }
38 cout<<"No";
39 }
40 return 0;
41 }
42
43
44
45

```

Input	Expected	Got
✓ 1 0 16 17 32 4 2 7 18 19 30 346	16 32	16 32 ✓
✓ 1 6 1 1 3 4 5 6 2 1 6	1 6	1 6 ✓

Passed all testcases ✓

**Success**

Marks for this submission: 1.00/1.00

0000000000

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

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

Status: Finished

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

Time taken: 47 sec

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  $A[j] - A[i] = k$ ,  $1 \leq j$

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:



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

Started on: Tuesday, 21 October 2025, 1:56 PM

State: Finished

Completed on: Tuesday, 21 October 2025, 1:57 PM

Time taken: 27 secs

Marks: 1.00/1.00

Grade: 88.88 out of 88 (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 37

6 2 7 10 15 37 246

Output:

10 37

Input:

1

3

6

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

6 1 2 3 4 5 6

2 1 8

Output:

1 6

For example:

Input	Result
1	10 37
3 10 17 37	
6	
6 2 7 10 15 37 246	

Answer: (partially correct) 0.50

```

1 // C++ implementation
2 // of the above approach
3 #include <iostream>
4 using namespace std;
5 int main() {
6     int T;
7     cin >> T;
8     while (T--) {
9         int N1, N2;
10        cin >> N1 >> N2;
11        int A[N1], B[N2];
12        for (int i = 0; i < N1; i++) {
13            cin >> A[i];
14        }
15        for (int i = 0; i < N2; i++) {
16            cin >> B[i];
17        }
18        int i = 0, j = 0;
19        int last = -1;
20        while (i < N1 && j < N2) {
21            if (A[i] == B[j]) {
22                if (A[i] != last) {
23                    cout << A[i] << " ";
24                    last = A[i];
25                }
26                i++;
27                j++;
28            } else if (A[i] < B[j]) {
29                i++;
30            } else {
31                j++;
32            }
33        }
34        cout << endl;
35    }
36    return 0;
37 }

```

```

20     }
21
22     vector<int> ans;
23     for (int i = 0; i < A2.size(); i++) {
24         vector<int> ans;
25         for (int j = 0; j < A1.size(); j++) {
26             if (A1[j] == A2[i]) {
27                 if (A1[j] != last_printed) {
28                     printAns.push_back(A1[j]);
29                     last_printed = A1[j];
30                 }
31             }
32             else if (A1[j] < A2[i]) {
33                 last++;
34             }
35             else {
36                 last++;
37             }
38         }
39         printAns.push_back(A2[i]);
40     }
41     return ans;
42 }
43

```

Input	Expected	Got
1	10 17	10 17 ✓
0 10 17 17		
0		
2 7 10 15 17 200		
1	1 0	1 0 ✓
0 1 2 2 3 4 5 6		
2		
1 0		

Passed all tests: ✓

Correct

Wrote for 215 submission 1.201.02

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

Started on: Tuesday, 31 October 2023, 1:56 PM

State: Finished

Completed on: Tuesday, 31 October 2023, 1:57 PM

Time taken: 37 sec

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

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

Status: Finished

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

Time taken: 1 min 2 secs

Marks: 1.00/1.00

Grade: 36.88 out of 38.00 (97%)

Question 1 : (Correct) Mark 1.00 out of 1.00 100% 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

5 10 17 17

6 2 7 10 15 17 246

Output:

10 17

Input:

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

6 1 2 3 4 5 6

2 1 6

Output:

1 6

For example:

Input	Result
1	10 17
2 10 17 17	
6	
2 7 10 15 17 246	

Answer: (penalty: 0.00)

```

1. #include <stdio.h>
2.
3. #define MAX 10000
4.
5. int main() {
6.     int T;
7.     scanf("%d", &T);
8.
9.     while (T-- > 0) {
10.         int N1, N2;
11.         scanf("%d %d", &N1, &N2);
12.
13.         int A1[MAX], A2[MAX];
14.         for (int i = 0; i < N1; i++) {
15.             scanf("%d", &A1[i]);
16.         }
17.
18.         for (int i = 0; i < N2; i++) {
19.             scanf("%d", &A2[i]);
20.         }
21.
22.         int i = 0, j = 0, last_printed = -1000000000;
23.         while (i < N1 && j < N2) {
24.             if (A1[i] == A2[j]) {
25.                 if (A1[i] != last_printed) {
26.                     printf("%d ", A1[i]);
27.                     last_printed = A1[i];
28.                 }
29.                 i++;
30.                 j++;
31.             } else if (A1[i] < A2[j]) {
32.                 i++;
33.             } else {
34.                 j++;
35.             }
36.         }
37.         printf("\n");
38.     }
39.     return 0;
40. }
```

## 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 5 Correct Mark 1.00 out of 1.00 P Flag question

Find Duplicates 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

A Line : n Elements

Output Format:

Elements x - that is repeated

For example:

Input	Result
5 1 2 3 3 4	3

Answer: (usually require 1 file)

```
1 #include <iostream>
2
3 #define MAX 10000
4
5 int main() {
6     int n;
7     scanf("%d", &n);
8     int freq[MAX] = {0};
9     int a;
```

Answer: (usually require 1 file)

```
1 #include <iostream>
2
3 #define MAX 10000
4
5 int main() {
6     int n;
7     scanf("%d", &n);
8     int freq[MAX] = {0};
9     int a;
10    for (int i = 0; i < n; i++) {
11        scanf("%d", &a);
12        if (freq[a] == 1) {
13            printf("%d\n", a);
14            return 0;
15        }
16        freq[a]++;
17    }
18    return 0;
19 }
```

	Input	Expected	Got	
✓	15 10 5 7 6 5 1 2 3 8 4 9	5	5	✓
✓	5 1 2 3 4 4	4	4	✓
✓	8 1 1 2 2 4	1	1	✓

Passed all tests ✓

Correct

Thanks for the submission 1/20/25.

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

Started on: Tuesday, 21 October 2025, 3:51 PM

State: Finished

Completed on: Tuesday, 21 October 2025, 3:52 PM

Time taken: 42 sec

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 <iostream.h>
2
3 #define MAX 100000
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 }

```

1 1 2 3 4

Answer: (penalty regime: 0 %)

```

1 #include <iostream.h>
2
3 #define MAX 100000
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 }

```

Input	Expected	Got
11 10 9 7 8 0 1 2 3 8 4 7	7	7 ✓
5 1 2 2 4 4	4	4 ✓
5 1 1 2 3 4	1	1 ✓

Passed all tests ✓

Correct

Marks for this submission: 1.00/1.00