

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

Started on: Tuesday, 21 October 2023, 3:58 PM

State: Finished

Completed on: Tuesday, 21 October 2023, 3:58 PM

Time taken: 39 secs

Marks: 1.00/1.00

Grade: 4.00 out of 4.00 (100%)

Question 1 - Coder - Marks available: 1.00 [Edit question](#)Given an array A of sorted integers and another non-negative integer k. And if there exists [i] 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 - 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:

15, as 5 - 7 = 8

So Return 1.

For example:

Input: Result

2	1
1	2
3	4

Answer (partial register 0.0%)

0 - No pair exists.

```

3 Adaptive MAX 1000MB
4
5 int main()
6 {
7     int n;
8     int A[100];
9
10    scanf("%d", &n);
11    for (int i = 0; i < n; i++) {
12        scanf("%d", &A[i]);
13    }
14    scanf("%d", &k);
15    int i = 0, j = 1;
16
17    while (i < n && j < n) {
18        if (A[i] + k == A[j]) {
19            printf("YES\n");
20            if (i <= j)
21                return 1;
22            else if (i > j)
23                return 1;
24        }
25        j++;
26    }
27
28    printf("NO\n");
29    return 0;
30}
31

```

Input	Expected	Get
✓ 2 2 2 5 6	1 1 ✓	✓
✓ 12 1 4 6 8 12 14 16 18 21 25 3	1 1 ✓	✓
✓ 18 1 2 3 5 11 14 16 24 25 29 8	0 0 ✓	✓
✗ 18 1 2 3 7 13 14 15 16 24 25 18	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, 2:57 PM

Status: Finished

Completed on: Tuesday, 21 October 2025, 3:18 PM

Time taken: 41 secs

Marks: 1.00/1.00

Grade: 4.00 out of 4.00 (100%)

Question 1: Correct - Marks: 2.00 of 2.00 [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 $|A[i] - A[j]| \leq k$.

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 1 - 1 = 0

So Return 1.

For example:

Input: Result

1	1
1	2
2	0

```

1. #include <iostream.h>
2. #define MAX 100000
3.
4. int main()
5. {
6.     int n, k;
7.     int A[MAX];
8.
9.     scanf("%d %d", &n, &k);
10.    for (int i = 0; i < n; i++) {
11.        cin >> A[i];
12.    }
13.    sort(A, A+n);
14.
15.    int i = 0, j = 1;
16.
17.    while (i < n && j < n) {
18.        if (A[i] + k <= A[j]) {
19.            cout << "YES" << endl;
20.            return 0;
21.        } else if (A[i] - A[j] > k) {
22.            j++;
23.        } else {
24.            i++;
25.        }
26.    }
27.
28.    cout << "NO" << endl;
29.    return 0;
30. }
```

Input	Expected	Get
✓ 2 1 3 5 4	1 1 ✓	1 ✓
✓ 20 1 4 6 8 12 14 15 18 21 25 1	1 1 ✓	1 ✓
✓ 20 1 2 3 5 11 14 15 16 18 20 0	0 0 ✓	0 ✓
✓ 20 0 2 3 7 13 14 15 18 19 20 10	1 1 ✓	1 ✓

```

26     scanf("%d", &M[0]);
27
28     int i = 0, j = 0;
29     int ans = -1000000000;
30
31     while (i < n1 && j < n2) {
32         if (M[i] == M[j]) {
33             if (ans <= M[i] - k)
34                 ans = M[i] - k;
35             last = A[i];
36         }
37         i++;
38         if (last <= M[i])
39             j++;
40     }
41     printf("%d\n";
42
43     return ans;
44 }
```

Input	Expected	Got
1 2 16 17 32 4 2 7 14 19 37 248	16 37 16 37 ✓	
1 0 1 2 3 4 5 6 2 3 6	1 6 1 6 ✓	

Passed all tests! ✓

Analysis

Marks for this submission: 1.00/1.00.

View code

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

Started on: Tuesday, 21 October 2023, 3:15 PM

Status: Pending

Completed on: Tuesday, 21 October 2023, 3:18 PM

Time taken: 47 secs

Marks: 1.00/1.00

Grade: 4.00 out of 4.00 (100%)

Question 1 (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 $A[j] - A[i] \geq k$. If yes, print 1 else print 0.

Input Format:

First Line n - Number of elements in an array

Next n Lines - n elements in the array

n - 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 2020, 1:56 PM

Status: Finished

Completed on: Tuesday, 21 October 2020, 1:57 PM

Time taken: 37 secs

Marks: 1.00/1.00

Grade: 88.00 out of 88.00 (100%)

Question 1

Correct Mark 1.00 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 N_1 , followed by N_1 integers of the first array.

2. Line 2 contains N_2 , followed by N_2 integers of the second array.

Output Format:

The intersection of the arrays on a single line.

Example:

Input:

1

3 10 17 19

4 2 7 10 15 17 24

Output:

10 17

Input:

1

6 123456

2 3 6

Output:

1 6

For example:

Input	Result
1	10 17
3 10 17 19	
4 2 7 10 15 17 24	

Answer: (marking original 2.0)

```

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

```
16    }
17    scnf("%d", &N2);
18    for (int i = 0; i < N2; i++) {
19        scnf("%d", &B[i]);
20    }
21}
22int i = 0, j = 0, last_printed = -100000000;
23while (C < N0 || D < N2) {
24    if (A[i] <= C) {
25        if (A[i] == last_printed) {
26            printf("%d", A[i]);
27            last_printed = A[i];
28        }
29        i++;
30    }
31    if (A[i] <= D) {
32        if (A[i] == last_printed) {
33            last_printed = A[i];
34        }
35        j++;
36    }
37}
38printf("\n");
39}
40}
41}
42}
43}
```

Input	Expected	Get
1 3 18 17 12 4 2 7 18 13 17 246	18 17 18 17 12	18 17 12
1 4 12 2 1 2 5 2 1 6	1 6 1 6	1 6

Passed all testcases!

ONLINE

Score 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, 1:54 PM

Status: Finished

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

Time taken: 37 secs

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

Started on: Tuesday, 21 October 2020, 3:53 PM

Status: Finished

Completed on: Tuesday, 21 October 2020, 3:55 PM

Time taken: 1 min 2 sec

Marks: 1.00/1.00

Grade: 88.88 out of 100.00 (88%)

Question 1 [Solved] | Mark 1.00 out of 1.00 | [Edit 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:

- 1. 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 12 23 45 56

2 1 6

Output:

15

For example:

Input	Result
1	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (Initially register: 0 %)

```

1. #include <iostream>
2.
3. #define MAX 100000
4.
5. int main()
6. {
7.     int N1, N2;
8.     cin >> N1 >> N2;
9.     vector<int> A(N1), B(N2);
10.    while (T--) {
11.        cin >> N1;
12.        cin >> N2;
13.        cin >> A[0];
14.        for (int i = 1; i < N1; i++) {
15.            cin >> A[i];
16.        }
17.        cin >> B[0];
18.        for (int j = 1; j < N2; j++) {
19.            cin >> B[j];
20.        }
21.
22.        int i = 0, j = 0, last_primed = -100000000;
23.        while (i < N1 && j < N2) {
24.            if (A[i] == B[j]) {
25.                if (A[i] != last_primed) {
26.                    cout << A[i] << " ";
27.                    last_primed = A[i];
28.                }
29.            }
30.            i++;
31.            j++;
32.        } while (i < N1 || j < N2) {
33.            cout << " ";
34.        }
35.    }
36. }
```

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

Started on: Tuesday, 21 October 2020, 3:54 PM

State: Finished

Completed on: Tuesday, 21 October 2020, 3:55 PM

Time taken: 29 secs

Marks: 1.00/1.00

Grade: 4.00 out of 4.00 (100%)

Question 1 / Correct More 1.00 out of 1.00 [Flag as incorrect](#)

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

A Line - n Elements

Output Format:

Element x - What is repeated

For example:

Input	Result
5 1 2 2 3 4	2

Answer: (Penalty regime: 0 %)

```

1. #include <iostream.h>
2. #define MAX 10000
3.
4. int main() {
5.     int n;
6.     cout<>"N:"<>n;
7.     int arr[MAX] = {0};
8.     int i;
9.     cin<>i;
10. }
```

Answer: (Penalty regime: 0 %)

```

1. #include <iostream.h>
2. #define MAX 10000
3.
4. int main() {
5.     int n;
6.     cout<>"N:"<>n;
7.     int arr[MAX] = {0};
8.     int i;
9.     for(i=0; i < n; i++) {
10.         cin<>x;
11.         if(x == arr[i]) {
12.             cout<>"Repeating:"<>x;
13.             return 0;
14.         }
15.         arr[x]++;
16.     }
17.     return 0;
18. }
```

Input	Expected	Get
11 2 3 7 4 5 1 2 3 8 6 7	1	1 ✓
1 1 2 3 4 4	4	4 ✓
3 1 2 2 3 4	1	1 ✓

Passed all test! ✓

[Correct](#)

Marked for review submission: 1.00/1.00

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 | Details | View 1.00 out of 1.00 | [Edit question](#)

Find Duplicate in Array:
Given a read-only array of 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 %)

```
#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    int freq[10000] = {0};
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        if (freq[x] == 1) {
            cout << x;
            return 0;
        }
        freq[x]++;
    }
    return 0;
}
```

Assignment: (penalty regime: 0 %)

```
#include <iostream>
using namespace std;
const int MAX = 10000;
int maxO () {
    int n;
    cout << "N: ";
    cin >> n;
    int freq[MAX + 10];
    int ans;
    for (int i = 0; i < n; i++) {
        int x;
        cin >> x;
        if (freq[x] == 1) {
            cout << x;
            cout << endl;
            freq[x] = 2;
        } else {
            cout << "Duplicated: " << x;
            freq[x]++;
        }
    }
    return 0;
}
```

Input **Expected** **Get**

11 1 1 2 3 1 2 2 8 4 7	1	1 ✓
5 1 2 2 4 4	4	4 ✓
5 1 1 2 3 4	1	1 ✓

Passed all testcases! ✓

CORRECT
Marks for this submission 1.00/1.00