

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

Started on	Sunday, 10 November 2024, 7:33 PM
State	Finished
Completed on	Wednesday, 20 November 2024, 10:30 AM
Time taken	9 days 14 hours
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 findPairWithDifference(int A[], int n, int k) {
4      int i = 0, j = 0;
5
6      while (j < n) {
7          int diff = A[j] - A[i];
8          if (diff == k && i != j) {
9              return 1;
10         } else if (diff < k) {
11             j++;
12         } else {
13             i++;
14             if (i == j) {
15                 j++;
16             }
17         }
18     }
19     return 0;
20 }
21
22 int main() {
23     int n;
24     scanf("%d", &n);
25     int A[n];
26
27     for (int i = 0; i < n; i++) {
28         scanf("%d", &A[i]);
29     }
30
31     int k;
32     scanf("%d", &k);
33

```

```
34     int result = findPairWithDifference(A, n, k);
35     printf("%d\n", result);
36
37     return 0;
38 }
```

	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 ▶