RAJALAKSHMI ENGINEERING COLLEGE

6

KAVIYA B J 2024-CSE ⌄

K2

Started on	Friday, 31 October 2025, 4:54 AM
State	Finished
Completed on	Friday, 31 October 2025, 4:56 AM
Time taken	2 mins
Marks	1.00/1.00
Grade	**4.00** out of 4.00 (**100**%)

6

KAVIYA B J 2024-CSE ⌄

K2

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 %)

```c
1   #include <stdio.h>
2
3   int main() {
4       int n;
5       scanf("%d", &n);
6       int arr[n];
7
8       for (int i = 0; i < n; i++) {
9           scanf("%d", &arr[i]);
10      }
11
12      // Using frequency array
13      int freq[n + 1];
14      for (int i = 0; i <= n; i++)
15          freq[i] = 0;
16
17      int duplicate = -1;
18
19      for (int i = 0; i < n; i++) {
20          if (freq[arr[i]] == 1) {
21              duplicate = arr[i];
22              break;
23          }
24          freq[arr[i]]++;
25      }
26
27      printf("%d\n", duplicate);
28      return 0;
29  }
30
```

	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.

Back to Course

6

KAVIYA B J 2024-CSE ⌄

K2

Started on	Friday, 31 October 2025, 4:57 AM
State	Finished
Completed on	Friday, 31 October 2025, 4:58 AM
Time taken	59 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 %)

```c
1   #include <stdio.h>
2
3   int main() {
4       int n;
5       scanf("%d", &n);
6
7       int arr[n];
8       for (int i = 0; i < n; i++) {
9           scanf("%d", &arr[i]);
10      }
11
12      int freq[n + 1]; // since elements are from 1 to n
13      for (int i = 0; i <= n; i++) {
14          freq[i] = 0;
15      }
16
17      int duplicate = -1;
18
19      for (int i = 0; i < n; i++) {
20          if (freq[arr[i]] == 1) {
21              duplicate = arr[i];
22              break;
23          }
24          freq[arr[i]]++;
25      }
26
27      printf("%d", duplicate);
28
29      return 0;
30  }
31
```

	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.

Back to Course

RAJALAKSHMI ENGINEERING COLLEGE

KAVIYA B J 2024-CSE ⌄   **K2**

Started on	Friday, 31 October 2025, 4:58 AM
State	Finished
Completed on	Friday, 31 October 2025, 4:59 AM
Time taken	1 min 11 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 ▾ int main() {
 4       int T;
 5       scanf("%d", &T);
 6
 7 ▾     while (T--) {
 8           int n1, n2;
 9           scanf("%d", &n1);
10           int a[n1];
11           for (int i = 0; i < n1; i++)
12               scanf("%d", &a[i]);
13
14           scanf("%d", &n2);
15           int b[n2];
16           for (int i = 0; i < n2; i++)
17               scanf("%d", &b[i]);
18
19           int i = 0, j = 0;
20           int found = 0;
21
22 ▾         while (i < n1 && j < n2) {
```

```
22
23 ▾            if (a[i] == b[j]) {
24                printf("%d ", a[i]);
25                found = 1;
26                i++;
27                j++;
28            }
29            else if (a[i] < b[j])
30                i++;
31            else
32                j++;
33        }
34
35        if (!found)
36            printf("No Intersection");
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.

Back to Course

RAJALAKSHMI
ENGINEERING
COLLEGE

6
KAVIYA B J 2024-CSE ˅
K2

Started on	Friday, 31 October 2025, 5:00 AM
State	Finished
Completed on	Friday, 31 October 2025, 5:01 AM
Time taken	1 min 3 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   int main() {
4       int T;
5       scanf("%d", &T);
6
7       while (T--) {
8           int n1, n2;
9           scanf("%d", &n1);
10          int a[n1];
11          for (int i = 0; i < n1; i++)
12              scanf("%d", &a[i]);
13
14          scanf("%d", &n2);
15          int b[n2];
16          for (int i = 0; i < n2; i++)
17              scanf("%d", &b[i]);
18
19          int i = 0, j = 0;
20          int found = 0;
21
22          // Two-pointer approach
```

```
22                    // Two pointer approach
23 ▾        while (i < n1 && j < n2) {
24 ▾            if (a[i] == b[j]) {
25                    printf("%d ", a[i]);
26                    found = 1;
27                    i++;
28                    j++;
29                }
30                else if (a[i] < b[j])
31                    i++;
32                else
33                    j++;
34            }
35
36            if (!found)
37                printf("No Intersection");
38
39            printf("\n");
40        }
41
42        return 0;
43 }
44
```

	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.

Back to Course

**RAJALAKSHMI
ENGINEERING
COLLEGE**

6

KAVIYA B J 2024-CSE ˅

K2

Started on	Friday, 31 October 2025, 5:01 AM
State	Finished
Completed on	Friday, 31 October 2025, 5:02 AM
Time taken	43 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 != 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 main() {
 4      int n;
 5      scanf("%d", &n);
 6
 7      int A[n];
 8      for (int i = 0; i < n; i++)
 9          scanf("%d", &A[i]);
10
11      int k;
12      scanf("%d", &k);
13
14      int i = 0, j = 1;
15      int found = 0;
16
17      // Two-pointer approach since array is sorted
18      while (i < n && j < n) {
19          if (i != j && A[j] - A[i] == k) {
20              found = 1;
21              break;
22          }
23          else if (A[j] - A[i] < k)
24              j++;
25          else
26              i++;
27      }
28
29      printf("%d", found);
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

Marks for this submission: 1.00/1.00.

Back to Course

KAVIYA B J 2024-CSE

K2

Started on	Friday, 31 October 2025, 5:02 AM
State	Finished
Completed on	Friday, 31 October 2025, 5:03 AM
Time taken	49 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 != 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 main() {
4       int n;
5       scanf("%d", &n);
6
7       int A[n];
8       for (int i = 0; i < n; i++)
9           scanf("%d", &A[i]);
10
11      int k;
12      scanf("%d", &k);
13
14      int i = 0, j = 1;
15      int found = 0;
16
17      // Two-pointer approach
18 ▾    while (i < n && j < n) {
19 ▾        if (i != j && A[j] - A[i] == k) {
20              found = 1;
21              break;
22          }
23          else if (A[j] - A[i] < k)
24              j++;
25          else
26              i++;
27      }
28
29      printf("%d", found);
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

Marks for this submission: 1.00/1.00.

Back to Course