<u>Dashbo</u>... / <u>My cour</u>... / <u>CS23331-DAA-2023-</u>... / <u>Competitive Program</u>... / <u>1-Finding Duplicates-O(n^2) Time Complexity,O(1) Space Co</u>...

Started on	Monday, 4 November 2024, 6:37 PM
State	Finished
Completed on	Monday, 4 November 2024, 6:54 PM
Time taken	16 mins 47 secs
Marks	1.00/1.00
Grade	4.00 out of 4.00 (100 %)

Question ${\bf 1}$

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 1 2 3 4	

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
 3 ▼ int main() {
         int n;
scanf("%d", &n);
 4
 5
 6
 7
         int num[n];
 8
         for (int i = 0; i < n; i++) {
              scanf("%d", &num[i]);
 9
10
         }
11
         for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {</pre>
12 🔻
13 🔻
14 🔻
                   if (num[j] == num[i]) {
                       printf("%d\n", num[i]);
15
16
                       return 0;
17
                   }
18
              }
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.

◄ 4-DP-Longest non-decreasing Subsequence

Jump to...

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

Started on	Monday, 4 November 2024, 6:54 PM	4
State	Finished	
Completed on	Monday, 4 November 2024, 6:59 PM	1
Time taken	5 mins 7 secs	
Marks	1.00/1.00	
Grade	4.00 out of 4.00 (100 %)	

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 1 2 3 4	

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
 2 v int main() {
 3
        int n;
 4
        scanf("%d", &n);
 5
        int arr[n];
        for (int i = 0; i < n; i++) {
 6 ▼
            scanf("%d", &arr[i]);
 7
 8
        }
9
        int sum = 0;
10
        int expected_sum = (n - 1) * n / 2;
        for (int i = 0; i < n; i++) {
11 •
12
            sum += arr[i];
13
14
        int duplicate = sum - expected_sum;
15
        printf("%d", duplicate);
16
        return 0;
17
18
```

	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.

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

Jump to...

<u>Dashb</u>... / <u>My cou</u>... / <u>CS23331-DAA-202</u>... / <u>Competitive Progra</u>... / <u>3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) S</u>...

Started on	Monday, 4 November 2024, 6:56 PM
State	Finished
Completed on	Monday, 4 November 2024, 7:12 PM
Time taken	15 mins 37 secs
Marks	1.00/1.00
C I.	20.00 - 1 - (.20.00 (4000))

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

6123456

2 1 6

Output:

16

For example:

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

```
#include <stdio.h>
 3 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
 4
         int i = 0, j = 0;
         int found = 0;
 5
 6
 7 •
         while (i < n1 && j < n2) {</pre>
             if (arr1[i] == arr2[j]) {
    printf("%d ", arr1[i]);
 8 ,
 9
10
                  i++;
11
                  j++;
12
                  found = 1;
             } else if (arr1[i] < arr2[j]) {</pre>
13 •
14
                  i++;
15 •
             } else {
16
                  j++;
17
18
         }
19
20
         if (!found) {
             printf("No common elements");
21
22
```

```
printt( \n );
23
24
25
26 v int main() {
27
          int T;
scanf("%d", &T);
28
29
          while (T--) {
30
               int n1, n2;
scanf("%d", &n1);
31
32
                int arr1[n1];
33
                for (int i = 0; i < n1; i++) {
    scanf("%d", &arr1[i]);</pre>
34 -
35
36
37
                scanf("%d", &n2);
38
                int arr2[n2];
                for (int i = 0; i < n2; i++) {
    scanf("%d", &arr2[i]);</pre>
39
40
41
42
43
                findIntersection(arr1, n1, arr2, n2);
44
45
46
          return 0;
47
     }
48
```

	Input	Expected	Got	
~	1	10 57	10 57	~
	3 10 17 57			
	6			
	2 7 10 15 57 246			
~	1	1 6	1 6	~
	6 1 2 3 4 5 6			
	2			
	1 6			

Correct

Marks for this submission: 1.00/1.00.

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

Jump to...

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

1.

Dashb... / My cou... / CS23331-DAA-202... / Competitive Progra... / 4-Print Intersection of 2 sorted arrays-O(m+n)Time Complexity,O(1) S...

Started on	Monday, 4 November 2024, 7:00 PM
State	Finished
Completed on	Monday, 4 November 2024, 7:12 PM
Time taken	11 mins 28 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

6123456

2 1 6

Output:

16

For example:

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

```
#include <stdio.h>
 3 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
 4
         int i = 0, j = 0;
         int found = 0;
 5
 6
 7 •
         while (i < n1 && j < n2) {</pre>
             if (arr1[i] == arr2[j]) {
    printf("%d ", arr1[i]);
 8 ,
 9
10
                  i++;
11
                  j++;
12
                  found = 1;
             } else if (arr1[i] < arr2[j]) {</pre>
13 •
14
                  i++;
15 •
             } else {
16
                  j++;
17
18
         }
19
20
         if (!found) {
             printf("No common elements");
21
22
```

```
printt( \n );
23
24
25
26 v int main() {
27
         int T;
scanf("%d", &T);
28
29
        while (T--) {
30
             int n1, n2;
scanf("%d", &n1);
31
32
             int arr1[n1];
33
             for (int i = 0; i < n1; i++) {</pre>
34
35
                 scanf("%d", &arr1[i]);
36
37
             scanf("%d", &n2);
38
39
             int arr2[n2];
             for (int i = 0; i < n2; i++) {
40
41
                 scanf("%d", &arr2[i]);
42
43
             findIntersection(arr1, n1, arr2, n2);
44
45
         }
46
47
         return 0;
48
49
```

	Input	Expected	Got	
~	1 3 10 17 57	10 57	10 57	~
	6 2 7 10 15 57 246			
~	1 6 1 2 3 4 5 6 2 1 6	1 6	1 6	~

Correct

Marks for this submission: 1.00/1.00.

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

Jump to...

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

1.

<u>Dashbo</u>... / <u>My cour</u>... / <u>CS23331-DAA-2023-</u>... / <u>Competitive Program</u>... / <u>5-Pair with Difference-O(n^2)Time Complexity,O(1) Space Co</u>...

Started on	Monday, 4 November 2024, 7:01 PM
State	Finished
Completed on	Monday, 4 November 2024, 7:11 PM
Time taken	10 mins 4 secs
Marks	1.00/1.00
C d	4.00 - 1 - 5.4.00 (4000)

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
1 3 5	
4	

```
#include <stdio.h>
 2
 3 | int findPairWithDifference(int arr[], int n, int k) {
 4
        int i = 0, j = 1;
 5
        while (j < n) {
 6
            int diff = arr[j] - arr[i];
 7
 8
 9 ,
             if (diff == k && i != j) {
10
                 return 1;
11 •
             } else if (diff < k) {</pre>
12
                 j++;
13
             } else {
14
                 i++;
15
16
17
             if (i == j) {
18
                 j++;
19
20
        }
21
        return 0;
22
23
    }
24
25
    int main() {
        int n, k;
scanf("%d", &n);
26
27
        int arr[n];
28
29
        for (int i = 0; i < n; i++) {
             scanf("%d", &arr[i]);
30
31
        scanf("%d", &k);
32
        int result = findPairWithDifference(arr, n, k);
33
34
        printf("%d\n", result);
35
36
        return 0;
37
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	•

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 ►

<u>Dashbo</u>... / <u>My cour</u>... / <u>CS23331-DAA-2023-A</u>... / <u>Competitive Program</u>... / <u>6-Pair with Difference -O(n) Time Complexity,O(1) Space Com</u>...

Started on	Monday, 4 November 2024, 7:03 PM
State	Finished
Completed on	Monday, 4 November 2024, 7:11 PM
Time taken	8 mins 28 secs
Marks	1.00/1.00
C	4.00 - 1 - 5.4.00 (4000)

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
1 3 5	
4	

```
#include <stdio.h>
 2
 3 | int findPairWithDifference(int arr[], int n, int k) {
 4
        int i = 0, j = 1;
 5
        while (j < n) {
 6
            int diff = arr[j] - arr[i];
 7
 8
 9 ,
             if (diff == k && i != j) {
10
                 return 1;
11 •
             } else if (diff < k) {</pre>
12
                 j++;
13
             } else {
14
                 i++;
15
16
17
             if (i == j) {
18
                 j++;
19
20
        }
21
        return 0;
22
23
    }
24
25
    int main() {
        int n, k;
scanf("%d", &n);
26
27
        int arr[n];
28
29
        for (int i = 0; i < n; i++) {
             scanf("%d", &arr[i]);
30
31
        scanf("%d", &k);
32
        int result = findPairWithDifference(arr, n, k);
33
34
        printf("%d\n", result);
35
36
        return 0;
37
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	~

CorrectMarks for this submission: 1.00/1.00.

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

Jump to...