<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 | Saturday, 2 November 2024, 1:35 PM |
|--------------|---|
| State | Finished |
| Completed on | Saturday, 2 November 2024, 1:37 PM |
| Time taken | 2 mins 53 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 1 2 3 4 | |

```
#include <stdio.h>
 3 | int find_duplicate(int* nums, int n) {
 4
        int slow = nums[0];
        int fast = nums[0];
 5
 6
 7
 8
        do {
 9
             slow = nums[slow];
10
            fast = nums[nums[fast]];
11
        } while (slow != fast);
12
13
14
        slow = nums[0];
15
        while (slow != fast) {
            slow = nums[slow];
16
17
            fast = nums[fast];
18
19
20
        return slow;
21
   }
22
23 v int main() {
24
        int n;
        scanf("%d", &n);
25
26
        int nums[n];
        for (int i = 0; i < n; i++) {
27
            scanf("%d", &nums[i]);
28
29
        }
30
31
        int result = find_duplicate(nums, n);
32
33
        printf("%d\n", result);
34
35
        return 0;
36
    }
37
```

| | 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

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

| Started on | Saturday, 2 November 2024, 2:15 PM |
|--------------|---|
| State | Finished |
| Completed on | Saturday, 2 November 2024, 2:17 PM |
| Time taken | 1 min 24 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 1 2 3 4 | |

```
#include <stdio.h>
 2
 3 | int find_duplicate(int* nums, int n) {
 4
        int slow = nums[0];
 5
        int fast = nums[0];
 6
 7
 8
        do {
 9
             slow = nums[slow];
10
            fast = nums[nums[fast]];
11
        } while (slow != fast);
12
13
14
        slow = nums[0];
15
        while (slow != fast) {
            slow = nums[slow];
16
17
            fast = nums[fast];
18
19
20
        return slow;
21
    }
22
23 v int main() {
24
        int n;
        scanf("%d", &n);
25
26
        int nums[n];
        for (int i = 0; i < n; i++) {
27
             scanf("%d", &nums[i]);
28
29
        }
30
31
        int result = find_duplicate(nums, n);
32
33
        printf("%d\n", result);
34
35
        return 0;
36
37
38
39
40
41
```

| | | 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! 🗸



Marks for this submission: 1.00/1.00.

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

Jump to...

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

Dashbo... / My cou... / CS23331-DAA-2023... / Competitive Progra... / 3-Print Intersection of 2 sorted arrays-O(m*n)Time Complexity,O(1) Sp...

| Started on | Saturday, 2 November 2024, 1:51 PM |
|--------------|---|
| State | Finished |
| Completed on | Saturday, 2 November 2024, 2:01 PM |
| Time taken | 9 mins 58 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:

.

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 3 10 17 57 | 10 57 |
| 6 | |
| 2 7 10 15 57 246 | |

```
int first = 1;
 6
 7 🔻
        while (i < n1 \&\& j < n2) {
 8 •
             if (arr1[i] < arr2[j]) {</pre>
 9
             } else if (arr1[i] > arr2[j]) {
10
11
                 j++;
             } else {
12
13
14
                 if (first) {
                     printf("%d", arr1[i]);
15
16
                     first = 0;
                 } else {
17
                     printf(" %d", arr1[i]);
18
19
20
                 i++;
```

```
11/17/24, 12:09 PM
```

```
21
                 j++;
22
23
        printf("\n");
24
25
26
27 v int main() {
28
        int T;
29
30
31
        scanf("%d", &T);
32
33
        while (T--) {
34
            int n1, n2;
35
36
            scanf("%d", &n1);
37
38
            int arr1[n1];
39
            for (int i = 0; i < n1; i++) {
40
                 scanf("%d", &arr1[i]);
41
42
43
44
            scanf("%d", &n2);
45
            int arr2[n2];
            for (int i = 0; i < n2; i++) {
46
                 scanf("%d", &arr2[i]);
47
48
            }
49
50
51
            find_intersection(arr1, n1, arr2, n2);
52
        }
53
54
        return 0;
55
    }
56
```

| | 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.

■ 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 ►

10

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 | Saturday, 2 November 2024, 2:09 PM |
|--------------|---|
| State | Finished |
| Completed on | Saturday, 2 November 2024, 2:12 PM |
| Time taken | 2 mins 58 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 3 10 17 57 | 10 57 |
| 2 7 10 15 57 246 | |

```
#include <stdio.h>
 3 void find_intersection(int arr1[], int n1, int arr2[], int n2) {
4
        int i = 0, j = 0;
 5
        int first = 1;
 6
        while (i < n1 \& j < n2) {
 7 🔻
 8 🔻
            if (arr1[i] < arr2[j]) {</pre>
9
                 i++;
10 •
            } else if (arr1[i] > arr2[j]) {
11
                j++;
12
            } else {
13
14
                 if (first) {
                     printf("%d", arr1[i]);
15
```

```
11/17/24, 12:10 PM
```

```
16
                     first = 0;
17
                 } else {
                     printf(" %d", arr1[i]);
18
19
                 }
20
                 i++;
21
                 j++;
22
             }
23
        }
24
        printf("\n");
25
    }
26
27 v int main() {
28
        int T;
29
         scanf("%d", &T);
30
31
32
        while (T--) {
33
             int n1, n2;
34
             scanf("%d", &n1);
35
             int arr1[n1];
             for (int i = 0; i < n1; i++) {
36
                 scanf("%d", &arr1[i]);
37
38
             scanf("%d", &n2);
39
             int arr2[n2];
40
41
             for (int i = 0; i < n2; i++) {</pre>
                 scanf("%d", &arr2[i]);
42
43
             find_intersection(arr1, n1, arr2, n2);
44
45
         }
46
47
        return 0;
48
    }
49
```

| | 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.

■ 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 | Saturday, 2 November 2024, 2:22 PM |
|--------------|---|
| State | Finished |
| Completed on | Saturday, 2 November 2024, 2:29 PM |
| Time taken | 6 mins 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
```

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 has_pair_with_difference(int A[], int n, int k) {
 4
        int i = 0, j = 0;
 5
 6
        while (i < n && j < n) {
 7
            int diff = A[j] - A[i];
 8
            if (diff == k && i != j) {
9 •
10
                 return 1;
11 •
             } else if (diff < k) {</pre>
12
                 j++;
13 🔻
             } else {
14
                i++;
15
16
        }
17
        return 0;
    }
18
19
20 v int main() {
21
        int n, k;
        scanf("%d", &n);
22
23
        int A[n];
24
        for (int i = 0; i < n; i++) {
25
            scanf("%d", &A[i]);
26
        }
27
        scanf("%d", &k);
        int result = has pair with difference(A, n, k);
28
        printf("%d\n", result);
29
30
31
        return 0;
32
33
```



| | 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...

11

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

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

| Started on | Saturday, 2 November 2024, 2:30 PM |
|--------------|---|
| State | Finished |
| Completed on | Saturday, 2 November 2024, 2:30 PM |
| Time taken | 53 secs |
| Marks | 1.00/1.00 |
| Grade | 4.00 out of 4.00 (100 %) |

118.185.187.137/moodle/mod/quiz/review.php?attempt=158652&cmid=1230

```
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 has_pair_with_difference(int A[], int n, int k) {
 4
        int i = 0, j = 0;
 5
 6
        while (i < n && j < n) {
 7
            int diff = A[j] - A[i];
 8
            if (diff == k && i != j) {
9 •
10
                 return 1;
11 •
             } else if (diff < k) {</pre>
12
                 j++;
13 🔻
             } else {
14
                i++;
15
16
        }
17
        return 0;
    }
18
19
20 v int main() {
21
        int n, k;
        scanf("%d", &n);
22
23
        int A[n];
24
        for (int i = 0; i < n; i++) {
25
            scanf("%d", &A[i]);
26
        }
27
        scanf("%d", &k);
        int result = has_pair_with_difference(A, n, k);
28
        printf("%d\n", result);
29
30
31
        return 0;
32
33
```

34

| | 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.

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

Jump to...