

Started on	Saturday, 16 November 2024, 8:02 PM
State	Finished
Completed on	Saturday, 16 November 2024, 8:05 PM
Time taken	3 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 2 3 4	1

Answer: (penalty regime: 0 %)

```

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

```

	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	✓

	Input	Expected	Got	
✓	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	Saturday, 16 November 2024, 8:06 PM
State	Finished
Completed on	Saturday, 16 November 2024, 8:10 PM
Time taken	4 mins 29 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 %)

```

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

```

	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	✓

	Input	Expected	Got	
✓	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...

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

Started on	Saturday, 16 November 2024, 8:51 PM
State	Finished
Completed on	Saturday, 16 November 2024, 8:59 PM
Time taken	7 mins 14 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 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
4     int i = 0, j = 0;
5     while (i < n1 && j < n2) {
6         if (arr1[i] < arr2[j]) {
7             i++;
8         } else if (arr1[i] > arr2[j]) {
9             j++;
10        } else {
11            printf("%d ", arr1[i]);
12            i++;
13            j++;
14        }
15    }
16    printf("\n");
17 }
18
19 int main() {
20     int t;
21
22     scanf("%d", &t);
23 }
```



```

23
24 while (t--) {
25     int n1, n2;
26     scanf("%d", &n1);
27     int arr1[n1];
28
29     for (int i = 0; i < n1; i++) {
30         scanf("%d", &arr1[i]);
31     }
32
33     scanf("%d", &n2);
34     int arr2[n2];
35
36     for (int i = 0; i < n2; i++) {
37         scanf("%d", &arr2[i]);
38     }
39     findIntersection(arr1, n1, arr2, n2);
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.



◀ 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 ▶

Started on	Saturday, 16 November 2024, 8:59 PM
State	Finished
Completed on	Saturday, 16 November 2024, 9:09 PM
Time taken	9 mins 48 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 void findIntersection(int arr1[], int n1, int arr2[], int n2) {
3     int i = 0, j = 0;
4     while (i < n1 && j < n2) {
5         if (arr1[i] < arr2[j]) {
6             i++;
7         } else if (arr1[i] > arr2[j]) {
8             j++;
9         } else {
10            printf("%d ", arr1[i]);
11            i++;
12            j++;
13        }
14    }
15    printf("\n");
16 }
17
18 int main() {
19     int t;
20     scanf("%d", &t);
21
22     while (t--) {
23         int n1, n2;
```

```

23     int n1, n2;
24     scanf("%d", &n1);
25     int arr1[n1];
26     for (int i = 0; i < n1; i++) {
27         scanf("%d", &arr1[i]);
28     }
29     scanf("%d", &n2);
30     int arr2[n2];
31     for (int i = 0; i < n2; i++) {
32         scanf("%d", &arr2[i]);
33     }
34
35     findIntersection(arr1, n1, arr2, n2);
36 }
37
38 return 0;
39 }
40

```

	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 ▶

Started on	Saturday, 16 November 2024, 9:09 PM
State	Finished
Completed on	Saturday, 16 November 2024, 9:15 PM
Time taken	5 mins 31 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 $A[j] - A[i] = k$, $i \neq j$.

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 int findPairWithDifference(int arr[], int n, int k) {
3     int i = 0, j = 1;
4     while (i < n && j < n) {
5         if (i != j && arr[j] - arr[i] == k) {
6             return 1;
7         } else if (arr[j] - arr[i] < k) {
8             j++;
9         } else {
10            i++;
11        }
12    }
13    return 0;
14 }
15
16 int main() {
17     int n, k;
18     scanf("%d", &n);
19     int arr[n];
20     for (int i = 0; i < n; i++) {
21         scanf("%d", &arr[i]);
22     }
23     scanf("%d", &k);
24     int result = findPairWithDifference(arr, n, k);
25     printf("%d\n", result);
26
27     return 0;
28 }
29

```

	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓

	Input	Expected	Got	
✓	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 ▶](#)

Started on	Saturday, 16 November 2024, 9:15 PM
State	Finished
Completed on	Saturday, 16 November 2024, 9:26 PM
Time taken	11 mins 2 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 $A[j] - A[i] = k$, $i \neq j$.

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  int findPairWithDifference(int arr[], int n, int k) {
3      int i = 0, j = 1;
4      while (i < n && j < n) {
5          if (i != j && arr[j] - arr[i] == k) {
6              return 1;
7          } else if (arr[j] - arr[i] < k) {
8              j++;
9          } else {
10             i++;
11         }
12     }
13
14     return 0;
15 }
16 int main() {
17     int n, k;
18     scanf("%d", &n);
19     int arr[n];
20     for (int i = 0; i < n; i++) {
21         scanf("%d", &arr[i]);
22     }
23     scanf("%d", &k);
24     int result = findPairWithDifference(arr, n, k);
25     printf("%d\n", result);
26     return 0;
27 }
28

```

	Input	Expected	Got	
✓	3 1 3 5 4	1	1	✓

	Input	Expected	Got	
✓	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...