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

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 2 3 4	1

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int find_duplicate(int* nums, int n) {
    int slow = nums[0];
    int fast = nums[0];


    do {
        slow = nums[slow];
        fast = nums[nums[fast]];
    } while (slow != fast);


    slow = nums[0];
    while (slow != fast) {
        slow = nums[slow];
        fast = nums[fast];
    }

    return slow;
}

int main() {
    int n;
    scanf("%d", &n);
    int nums[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }


    int result = find_duplicate(nums, n);
    printf("%d\n", result);

    return 0;
}

```

	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	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 2 3 4	1

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>

int find_duplicate(int* nums, int n) {
    int slow = nums[0];
    int fast = nums[0];


    do {
        slow = nums[slow];
        fast = nums[nums[fast]];
    } while (slow != fast);


    slow = nums[0];
    while (slow != fast) {
        slow = nums[slow];
        fast = nums[fast];
    }

    return slow;
}

int main() {
    int n;
    scanf("%d", &n);
    int nums[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &nums[i]);
    }


    int result = find_duplicate(nums, n);
    printf("%d\n", result);

    return 0;
}
```

	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...

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:

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 %)

```
    int first = 1;

    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr1[i] > arr2[j]) {
            j++;
        } else {

            if (first) {
                printf("%d", arr1[i]);
                first = 0;
            } else {
                printf(" %d", arr1[i]);
            }
            i++;
            j++;
        }
    }
    printf("\n");
}

int main() {
    int T;


    scanf("%d", &T);

    while (T--) {
        int n1, n2;


        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }


        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }


        find_intersection(arr1, n1, arr2, n2);
    }

    return 0;
}

```

	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 ►

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

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 %)

```c
#include <stdio.h>

void find_intersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int first = 1;

    while (i < n1 && j < n2) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr1[i] > arr2[j]) {
            j++;
        } else {

            if (first) {
                printf("%d", arr1[i]);
                first = 0;
            } else {
                printf(" %d", arr1[i]);
            }
            i++;
            j++;
        }
    }
    printf("\n");
}

int main() {
    int T;

    scanf("%d", &T);

    while (T--) {
        int n1, n2;
        scanf("%d", &n1);
        int arr1[n1];
        for (int i = 0; i < n1; i++) {
            scanf("%d", &arr1[i]);
        }
        scanf("%d", &n2);
        int arr2[n2];
        for (int i = 0; i < n2; i++) {
            scanf("%d", &arr2[i]);
        }
        find_intersection(arr1, n1, arr2, n2);
    }

    return 0;
}

```

	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 ►

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

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 3 5
4
	1

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int has_pair_with_difference(int A[], int n, int k) {
    int i = 0, j = 0;

    while (i < n && j < n) {
        int diff = A[j] - A[i];

        if (diff == k && i != j) {
            return 1;
        } else if (diff < k) {
            j++;
        } else {
            i++;
        }
    }
    return 0;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int A[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &A[i]);
    }
    scanf("%d", &k);
    int result = has_pair_with_difference(A, n, k);
    printf("%d\n", result);

    return 0;
}

```

	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 ►

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**%)

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 %)

```c
#include <stdio.h>

int has_pair_with_difference(int A[], int n, int k) {
    int i = 0, j = 0;

    while (i < n && j < n) {
        int diff = A[j] - A[i];

        if (diff == k && i != j) {
            return 1;
        } else if (diff < k) {
            j++;
        } else {
            i++;
        }
    }
    return 0;
}

int main() {
    int n, k;
    scanf("%d", &n);
    int A[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &A[i]);
    }
    scanf("%d", &k);
    int result = has_pair_with_difference(A, n, k);
    printf("%d\n", result);

    return 0;
}
```

	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...