| | |
|---|---|
| **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 **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<br>1 1 2 3 4 | 1 |

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

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

    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (num[j] == num[i]) {
                printf("%d\n", num[i]);
                return 0;
            }
        }
    }

    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 11<br>10 9 7 6 5 1 2 3 8 4 7 | 7 | 7 | ✔ |
| ✔ | 5<br>1 2 3 4 4 | 4 | 4 | ✔ |
| ✔ | 5<br>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 |
| **State** | Finished |
| **Completed on** | Monday, 4 November 2024, 6:59 PM |
| **Time taken** | 5 mins 7 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<br>1 1 2 3 4 | 1 |

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int sum = 0;
    int expected_sum = (n - 1) * n / 2;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    int duplicate = sum - expected_sum;
    printf("%d", duplicate);
    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 11<br>10 9 7 6 5 1 2 3 8 4 7 | 7 | 7 | ✔ |
| ✔ | 5<br>1 2 3 4 4 | 4 | 4 | ✔ |
| ✔ | 5<br>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** | 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 |
| **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<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57 |

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>

void findIntersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int found = 0;

    while (i < n1 && j < n2) {
        if (arr1[i] == arr2[j]) {
            printf("%d ", arr1[i]);
            i++;
            j++;
            found = 1;
        } else if (arr1[i] < arr2[j]) {
            i++;
        } else {
            j++;
        }
    }

    if (!found) {
        printf("No common elements");
    }
```

```
    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]);
        }

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

    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 1<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57 | 10 57 | ✔ |
| ✔ | 1<br>6 1 2 3 4 5 6<br>2<br>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** | 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

6 1 2 3 4 5 6

2 1 6

Output:

1 6

**For example:**

| Input | Result |
|---|---|
| 1<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57 |

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>

void findIntersection(int arr1[], int n1, int arr2[], int n2) {
    int i = 0, j = 0;
    int found = 0;

    while (i < n1 && j < n2) {
        if (arr1[i] == arr2[j]) {
            printf("%d ", arr1[i]);
            i++;
            j++;
            found = 1;
        } else if (arr1[i] < arr2[j]) {
            i++;
        } else {
            j++;
        }
    }

    if (!found) {
        printf("No common elements");
    }
```

```
    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]);
        }

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

    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 1<br>3 10 17 57<br>6<br>2 7 10 15 57 246 | 10 57 | 10 57 | ✔ |
| ✔ | 1<br>6 1 2 3 4 5 6<br>2<br>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** | 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 |
| **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<br>1 3 5<br>4 | 1 |

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>

int findPairWithDifference(int arr[], int n, int k) {
    int i = 0, j = 1;

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

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

        if (i == j) {
            j++;
        }
    }

    return 0;
}

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

    return 0;
}

```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 3<br>1 3 5<br>4 | 1 | 1 | ✔ |
| ✔ | 10<br>1 4 6 8 12 14 15 20 21 25<br>1 | 1 | 1 | ✔ |
| ✔ | 10<br>1 2 3 5 11 14 16 24 28 29<br>0 | 0 | 0 | ✔ |
| ✔ | 10<br>0 2 3 7 13 14 15 20 24 25<br>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** | 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 |
| **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<br>1 3 5<br>4 | 1 |

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>

int findPairWithDifference(int arr[], int n, int k) {
    int i = 0, j = 1;

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

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

        if (i == j) {
            j++;
        }
    }

    return 0;
}

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

    return 0;
}

```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✔ | 3<br>1 3 5<br>4 | 1 | 1 | ✔ |
| ✔ | 10<br>1 4 6 8 12 14 15 20 21 25<br>1 | 1 | 1 | ✔ |
| ✔ | 10<br>1 2 3 5 11 14 16 24 28 29<br>0 | 0 | 0 | ✔ |
| ✔ | 10<br>0 2 3 7 13 14 15 20 24 25<br>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...