Competitive programming

Roll no.: 241801152

Name: Manoj PT

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

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

```
1    #include <stdio.h>
2
3 ▾  int main() {
4        int n;
5        scanf("%d", &n);
6
7        int arr[n];
8 ▾      for (int i = 0; i < n; i++) {
9            scanf("%d", &arr[i]);
10       }
11 ▾     for (int i = 0; i < n; i++) {
12 ▾         for (int j = i + 1; j < n; j++) {
13 ▾             if (arr[i] == arr[j]) {
14                    printf("%d\n", arr[i]);
15                    return 0;
16                }
17            }
18        }
19
20        return 0;
21   }
22
```

	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.

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

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

```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]);
    }
    for(int i=0;i<n;i++){
        for(int j=i+1;j<n;j++){
            if(arr[i]==arr[j]){
                printf("%d ",arr[i]);
                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! ✔

Marks for this submission: 1.00/1.00.

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

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
1   #include<stdio.h>
2   int main(){
3       int n;
4       int t;
5       scanf("%d ",&t);
6       while(t--){
7           scanf("%d ",&n);
8           int arr[n];
9           for(int i=0;i<n;i++){
10              scanf("%d ",&arr[i]);
11          }
12          int m;
13          scanf("%d ",&m);
14          int arr1[m];
15
16          for(int i=0;i<m;i++){
17              scanf("%d ",&arr1[i]);
18          }
19          for(int i=0;i<n;i++){
20              for(int j=0;j<m;j++){
21                  if(arr[i]==arr1[j]){
22                      printf("%d ",arr[i]);
23                  }
24              }
25
26
27          }
28      }
29  }
```

	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.

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

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>
int main(){
    int n;
    int t;
    scanf("%d ",&t);
    while(t--){
        scanf("%d ",&n);
        int arr[n];
        for(int i=0;i<n;i++){
            scanf("%d ",&arr[i]);
        }
        int m;
        scanf("%d ",&m);
        int arr1[m];

        for(int i=0;i<m;i++){
            scanf("%d ",&arr1[i]);
        }
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                if(arr[i]==arr1[j]){
                    printf("%d ",arr[i]);
                }
            }
        }

    }
}
```

	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.

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

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

```c
1   #include <stdio.h>
2
3   int main() {
4       int n, k;
5       scanf("%d", &n);
6       int A[n];
7       for (int i = 0; i < n; i++) {
8           scanf("%d", &A[i]);
9       }
10      scanf("%d", &k);
11      int i = 0, j = 1, found = 0;
12      while (i < n && j < n) {
13          int diff = A[j] - A[i];
14          if (diff == k && i != j) {
15              found = 1;
16              break;
17          } else if (diff < k) {
18              j++;
19          } else {
20              i++;
21          }
22          if (i == j) {
23              j++;
24          }
25      }
26      printf("%d", found);
27      return 0;
28  }
29
```

	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.

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

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

```c
#include <stdio.h>
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 i = 0, j = 1, found = 0;
    while (i < n && j < n) {
        int diff = A[j] - A[i];
        if (diff == k && i != j) {
            found = 1;
            break;
        } else if (diff < k) {
            j++;
        } else {
            i++;
        }
        if (i == j) {
            j++;
        }
    }
    printf("%d", found);
    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! ✓