B.NIKITHA
230701211
CSE -'D'
III SEM

Competitive Programming

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

Answer:(penalty regime: 0 %)
```
#include<stdio.h>
int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    int b[n];
    for(int i=0;i<n;i++){
        b[i]=0;
    }

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

        }
        for(int i=0;i<n;i++){
            if(b[i]>1){
                printf("%d",i);
            }
        }
    }
```

Feedback

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

Answer:(penalty regime: 0 %)
```c
#include <stdio.h>

int findDuplicate(int* nums, int numsSize) {
    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 a[n];
   for(int i=0;i<n;i++){
       scanf("%d",&a[i]);
   }
   printf("%d",findDuplicate(a,n));


    return 0;
}
```

Feedback

```
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.

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 %)

```
#include<stdio.h>
int main(){
    int n,t;
    scanf("%d",&t);
    while(t>0){
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int m;
    scanf("%d",&m);
    int b[m];
    int d=m+n;
    int c[d];
    int k=0;
    for(int i=0;i<m;i++){
        scanf("%d",&b[i]);
        for(int j=0;j<n;j++){
            if(a[j]==b[i]){
            c[k]=b[i];
            k++;}
        }
    }
    for(int i=0;i<k;i++){
        printf("%d ",c[i]);
        c[i]=0;
    }
        t--;
        k=0;
    }
}
Feedback
Input            Expected     Got
1                10 57        10 57
3 10 17 57
6
2 7 10 15 57 246
1                1 6          1 6
6 1 2 3 4 5 6
2
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 %)

```
#include<stdio.h>
int main(){
    int t;
    scanf("%d",&t);
    while(t>0){
        int n;
        scanf("%d",&n);
        int a[n];
        for(int i=0;i<n;i++){
            scanf("%d",&a[i]);
        }
        int m;
        scanf("%d",&m);
        int b[m];
        for(int i=0;i<m;i++){
            scanf("%d",&b[i]);
        }
        int c[m+n];
        int k=0;
        int i=0,j=0;
        while(i<n&&j<m){
            if(a[i]>b[j]){
                j++;
            }
            else if(a[i]<b[j]){
                i++;
            }
            else {
                c[k]=a[i];
                i++;
                j++;
                printf("%d ",c[k]);
                k++;
            }
        }
        k=0;
        t--;
    }
}
```
Feedback

```
Input              Expected     Got
1                  10 57        10 57
3 10 17 57
6
2 7 10 15 57 246
1                  1 6          1 6
6 1 2 3 4 5 6
2
1 6
```

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
1 3 5
4
```
Answer:(penalty regime: 0 %)
```
#include<stdio.h>
#include<stdlib.h>
int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int k,f=0;
    scanf("%d",&k);
    for(int i=0;i<n;i++){
        for(int j=0;j<n-1;j++){
            if(abs(a[i]-a[j])==k && i!=j){
                f=1;
                printf("%d",1);
            }
            if(f==1){
                break;
            }
        }
    }
    if(f==0){
        printf("%d",0);
    }
}
```

Feedback

```
Input                    Expected  Got
3                        1         1
1 3 5
4
10                        1         1
1 4 6 8 12 14 15 20 21 25
1
10                        0         0
1 2 3 5 11 14 16 24 28 29
0
10                        1         1
0 2 3 7 13 14 15 20 24 25
10
```

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

Answer:(penalty regime: 0 %)

```
#include<stdio.h>
#include<stdlib.h>
int find(int a[],int n,int k){
     int i=0,j=1;
    while(i<n && j<n){
        if(abs(a[j]-a[i])<k)
        j++;
        else if(abs(a[j]-a[i])>k)
        i++;
        else if(abs(a[j]-a[i])==k && i!=j)
        return 1;
        if(i==j){
            j++;
        }
        }
        return 0;
    }


int main(){
    int n;
    scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int k;
    scanf("%d",&k);
    printf("%d",find(a,n,k));
    return 0;
}
```

Feedback

```
Input                        Expected  Got
3                            1         1
1 3 5
4
10                           1         1
1 4 6 8 12 14 15 20 21 25
1
10                           0         0
1 2 3 5 11 14 16 24 28 29
0
10                           1         1
0 2 3 7 13 14 15 20 24 25
10
```

Passed all tests!

Correct
Marks for this submission: 1.00/1.00.