	Input	Expected	Got	
✔	9 -1 3 4 5 2 2 2 2 3	6	6	✔
✔	7 1 2 2 4 5 7 6	6	6	✔

Passed all tests! ✔

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

program:

```
#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 flag=0;
    for(int i=0;i<n;i++){
        int el1=arr[i];

        for(int j=0;j<n;j++){
            if (el1==arr[j] && i!=j){
                printf("%d",el1);
                flag=1;
                break;
            }
        }
        if(flag)
        break;

    }
}
```

	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! ✔

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

program:

```
#include <stdio.h>
int main(){
   int n;
   scanf("%d",&n);
   int a[n];
   for(int i=0;i <n;i++){
      scanf("%d",&a[i]);
   }
   int b[n];
   for(int i=0;i <n;i++){
      b[i]=0;
   }
   for(int i=0;i<n;i++){
      //if el already present i.e, b[i]=1
      if(b[a[i]]){
         printf("%d",a[i]);
         break;
      }
      else
```

```
    b[a[i]]=1;
  }

}
```

	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! ✔

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

program:

```
#include<stdio.h>
int main(){
   int n;
   scanf("%d",&n);
   for(int i=0;i<n;i++){
      int n1;
      scanf("%d",&n1);
      int arr1[n1];
      for(int j=0;j<n1;j++){
         scanf("%d ",&arr1[j]);
      }
      int n2;
      scanf("%d",&n2);
      int arr2[n2];
```

```
    for(int j=0;j<n2;j++){
        scanf("%d ",&arr2[j]);
    }
    for(int j=0;j<n1;j++){
        for(int k=0;k<n2;k++){
            if(arr1[j]==arr2[k]){
                printf("%d ",arr1[j]);
            }
        }
    }
  }

}
```

	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	✔

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

program:

```
#include <stdio.h>

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]);
    }

    int i = 0, j = 0;
    while (i < n1 && j < n2) {
       if (arr1[i] < arr2[j]) {
          i++;
       }
       else if (arr2[j] < arr1[i]) {
          j++;
       }
       else {
          printf("%d ", arr1[i]);
          i++;
          j++;
       }
    }
    printf("\n");
  }

}
```

	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	✔

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.

program:

```
#include <stdio.h>
#include <stdlib.h>

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

```
int arr[n];



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

int t;
scanf("%d", &t);

int flag = 0;

for (int i = 0; i < n; i++) {
   for (int j = 0; j < n; j++) {
      if (i!=j && abs(arr[i] - arr[j]) == t) {
         flag = 1;
         break;
      }
   }
   if (flag) {
      break;
   }
}

if (flag) {
   printf("%d\n", 1);
} else {
   printf("%d\n", 0);
}

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! ✔

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.

program:

```
#include <stdio.h>
#include <stdlib.h>

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

    int arr[n];



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

    int t;
    scanf("%d", &t);

    int flag = 0;

    int i=0;
    int j=1;
    while(i<n && j<n){
        int diff = abs(arr[i] - arr[j]);
        if(i!=j && diff==t){
            flag=1;
            break;
        }
        else if(diff<t){
            j++;
        }
```

```
        else{
            i++;
        }

    }


    if (flag) {
        printf("%d\n", 1);
    } else {
        printf("%d\n", 0);
    }

    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! ✔