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 1 2 3 4	

Answer: (penalty regime: 0 %)

```
1 #include<stdio.h>
  2 v int main(){
      int a;
scanf("%d",&a);
  3
  4
       int arr[a];
  5
      int b = 0, c = 0;
for(int i=0;i<a;i++){
    scanf("%d",&arr[i]);
    i = confile</pre>
  6
  7 *
  8
  9
          c += arr[i];
 10
 11 v
       for(int i=1;i<a;i++){
 12
         b += i;
 13
 14
        int ans = c-b;
 15
        printf("%d",ans);
 16 }
```

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 1 2 3 4	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
 2 v int f(int arr[], int n) {
     int s = arr[0];
 3
     int fa= arr[0];
 4
5 v
     do {
       s = arr[s];
fa= arr[arr[fa]];
 6
8 } while (s != fa);
9 s= arr[0];
10 v while (s!= fa) {
       s= arr[s];
11
12
        fa= arr[fa];
13
14
    return s;
15
16 v int main() {
17
    int n;
18
    scanf("%d", &n);
19
    int arr[n];
20 v for (int i = 0; i < n; i++) {
21
       scanf("%d", &arr[i]);
22
    int d = f(arr, n);
23
     printf("%d\n", d);
24
25
      return 0;
26
27 }
```

# **COMPETITIVE PROGRAMMING**

6123456

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 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: 3 10 17 57 6 2 7 10 15 57 246 Output: 10 57 Input: 1

```
1 #include <stdio.h>
 2 v int main() {
              int T;
scanf("%d", &T);
 4
             while (T--) {
   int n1, n2;
   scanf("%d", &n1);
   int arr1[n1];
   for (int i = 0; i < n1; i++) {
      scanf("%d", &arr1[i]);
}</pre>
 6 ₹
10 v
12
14
                     scanf("%d", &n2);
int arr2[n2];
for (int i = 0; i < n2; i++) {
    scanf("%d", &arr2[i]);</pre>
17 v
18
20
                      while (i < n1 \&\& j < n2) {
                            if (arr1[i] == arr2[j]) {
   printf("%d ", arr1[i]);
                                    j++;
28
                             else if (arr1[i] < arr2[j]) {
30
                                   j++;
38
39
```

## **COMPETITIVE PROGRAMMING**

Output:

16

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: Line 1 contains N1, followed by N1 integers of the first array Line 2 contains N2, followed by N2 integers of the second array 2. **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 6123456 216

```
#include<stdio.h>
 2 v int main() {
              int T;
scanf("%d", &T);
                    int n1, n2;
scanf("%d", &n1);
int arr1[n1];
for (int i = 0; i < n1; i++) {
    scanf("%d", &arr1[i]);
}</pre>
                     scanf("%d", &n2);
int arr2[n2];
for (int i = 0; i < n2; i++) {
    scanf("%d", &arr2[i]);</pre>
17 v
                     int i = 0, j = 0;
20
                     while (i < n1 \&\& j < n2) {
                            if (arr1[i] == arr2[j]) {
    printf("%d ", arr1[i]);
24 ▼
                                   j++;
29 🔻
                            else if (arr1[i] < arr2[j]) {
30
32 √
                                   j++;
```

## **COMPETITIVE PROGRAMMING**

```
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
```

```
#include <stdio.h>
      int difference(int arr[], int n, int k) {
            int i = 0, j = 1;
while (j < n) {
   if (i != j && arr[j] - arr[i] == k)
     return 1;</pre>
 5 🔻
                   else if (arr[j] - arr[i] < k)
                          j++;
10
                          i++;
13
16 v
      int main() {
             int n, k;
scanf("%d", &n);
18
            int arr[n];
for (int i = 0; i < n; i++)
    scanf("%d", &arr[i]);
scanf("%d\n", difference(arr, n, k));</pre>
21
24
```

1 3 5 4

```
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.
```

```
#include <stdio.h>
 2
 3 v int difference(int arr[], int n, int k) {
         while (j < n) {
              if (i != j && arr[j] - arr[i] == k)
              else if (arr[j] - arr[i] < k)
 8
                   j++;
10
11
                   i++;
12
13
         return 0;
14
15
16 v int main() {
         int n, k;
scanf("%d", &n);
17
18
19
         int arr[n];
20
         for (int i = 0; i < n; i++)
         scanf("%d", &arr[i]);
scanf("%d", &k);
printf("%d\n",difference(arr, n, k));
21
22
24
         return 0;
25
```