

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 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int arr[n + 1];
    int freq[n + 1];
```

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);
    int arr[n + 1];
    int freq[n + 1];

    for (int i = 0; i <= n; i++)
        freq[i] = 0;

    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        freq[arr[i]]++;
        if (freq[arr[i]] > 1) {
            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	✓

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
3 int main() {
4     int n;
5     scanf("%d", &n);
6     int arr[n];
7     int seen[n + 1];
8     for (int i = 0; i <= n; i++)
9         seen[i] = 0;
10
11     for (int i = 0; i < n; i++) {
12         scanf("%d", &arr[i]);
13         if (seen[arr[i]]) {
14             printf("%d", arr[i]);
15             return 0;
16         }
```

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

	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	✓

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	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int t;
5     scanf("%d", &t);
6     while (t--) {
7         int n1, n2;
8         scanf("%d", &n1);
9         int a[n1];
10        for (int i = 0; i < n1; i++)
11            scanf("%d", &a[i]);
12
13        scanf("%d", &n2);
14        int b[n2];
15        for (int i = 0; i < n2; i++)
16            scanf("%d", &b[i]);
17
18        int i = 0, j = 0;
19        while (i < n1 && j < n2) {
20            if (a[i] == b[j]) {
21                printf("%d ", a[i]);
22                i++;
23                j++;
24            } else if (a[i] < b[j]) {
25                i++;
26            } else {
27                j++;
```

```

18     int i = 0, j = 0;
19     while (i < n1 && j < n2) {
20         if (a[i] == b[j]) {
21             printf("%d ", a[i]);
22             i++;
23             j++;
24         } else if (a[i] < b[j]) {
25             i++;
26         } else {
27             j++;
28         }
29     }
30     printf("\n");
31 }
32 return 0;
33 }
34

```

	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

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	10 57
3 10 17 57	
6	
2 7 10 15 57 246	

Answer: (penalty regime: 0 %)

```
1 #include <stdio.h>
2
3 int main() {
4     int T;
5     scanf("%d", &T);
6     while (T--) {
7         int n1, n2;
8         scanf("%d", &n1);
9         int a[n1];
10        for (int i = 0; i < n1; i++)
11            scanf("%d", &a[i]);
12
13        scanf("%d", &n2);
14        int b[n2];
15        for (int i = 0; i < n2; i++)
16            scanf("%d", &b[i]);
17
18        int i = 0, j = 0;
19        while (i < n1 && j < n2) {
20            if (a[i] == b[j]) {
21                printf("%d ", a[i]);
22                i++;
23                j++;
24            } else if (a[i] < b[j]) {
25                i++;
26            } else {
```

```

19 while (i < n1 && j < n2) {
20     if (a[i] == b[j]) {
21         printf("%d ", a[i]);
22         i++;
23         j++;
24     } else if (a[i] < b[j]) {
25         i++;
26     } else {
27         j++;
28     }
29 }
30 printf("\n");
31 }
32 return 0;
33 }
34

```

	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.

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

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 %)

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

	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.

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that $A[j] - A[i] = k$, $i \neq j$.

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	

Input	Result
3	1
1 3 5	
4	

Answer: (penalty regime: 0 %)

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

	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.