```c
#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 k;
    scanf("%d", &k);
    int i = 0, j = 1;
    int found = 0;
    while (i < n && j < n) {
        if (i != j) {
            int diff = A[j] - A[i];
            if (diff == k) {
                found = 1;
                break;
            } else if (diff < k) {
                j++;
            } else {
                i++;
            }
        } else {
            j++;
        }
    }
    printf("%d\n", found);
    return 0;
}
```

```c
#include <stdio.h>
int main() {
    int n;
    scanf("%d", &n);  // number of elements
    int A[n];
    for (int i = 0; i < n; i++) {
        scanf("%d", &A[i]);
    }
    int k;
    scanf("%d", &k);  // non-negative integer k
    int i = 0, j = 1; // two pointers
    int found = 0;
    while (i < n && j < n) {
        if (i != j && A[j] - A[i] == k) {
            found = 1;
            break;
        } else if (A[j] - A[i] < k) {
            j++;
        } else {
            i++;
        }
    }
    printf("%d\n", found);
    return 0;
}
```

urse index

	Input	Expected	Got	
✔	3 1 3 5 4	1	1	✔

```c
#include <stdio.h>

int main() {
    int T;
    scanf("%d", &T);  // number of test cases

    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;
        int printed = 0;

        // Two-pointer approach
        while (i < N1 && j < N2) {
            if (arr1[i] == arr2[j]) {
                if (printed) printf(" ");  // space between numbers
                printf("%d", arr1[i]);
                printed = 1;
                i++;
                j++;
            } else if (arr1[i] < arr2[j]) {
                i++;
            } else {
                j++;
            }
        }
        printf("\n");
    }
```

```c
#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;
        int first = 1;

        while (i < N1 && j < N2) {
            if (arr1[i] == arr2[j]) {
                if (!first) printf(" ");
                printf("%d", arr1[i]);
                first = 0;
                i++;
                j++;
            }
            else if (arr1[i] < arr2[j]) {
                i++;
            } else {
                j++;
            }
        }
        printf("\n");
    }
    return 0;
}
```

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

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

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

    int *visited = (int *)calloc(n + 1, sizeof(int)); // elements between 1 and n

    int duplicate = -1;
    for (int i = 0; i < n; i++) {
        if (visited[arr[i]] == 1) {
            duplicate = arr[i];
            break;
        }
        visited[arr[i]] = 1;
    }

    printf("%d\n", duplicate);

    free(arr);
    free(visited);
    return 0;
}
```

```c
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n;
    scanf("%d", &n);
    int *arr = (int *)malloc(n * sizeof(int));
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int *visited = (int *)calloc(n + 1, sizeof(int));  // n+1 because elements are 1 to n
    int duplicate = -1;
    for (int i = 0; i < n; i++) {
        if (visited[arr[i]] == 1) {
            duplicate = arr[i];
            break;
        }
        visited[arr[i]] = 1;
    }
    printf("%d\n", duplicate);
    free(arr);
    free(visited);
    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	✔