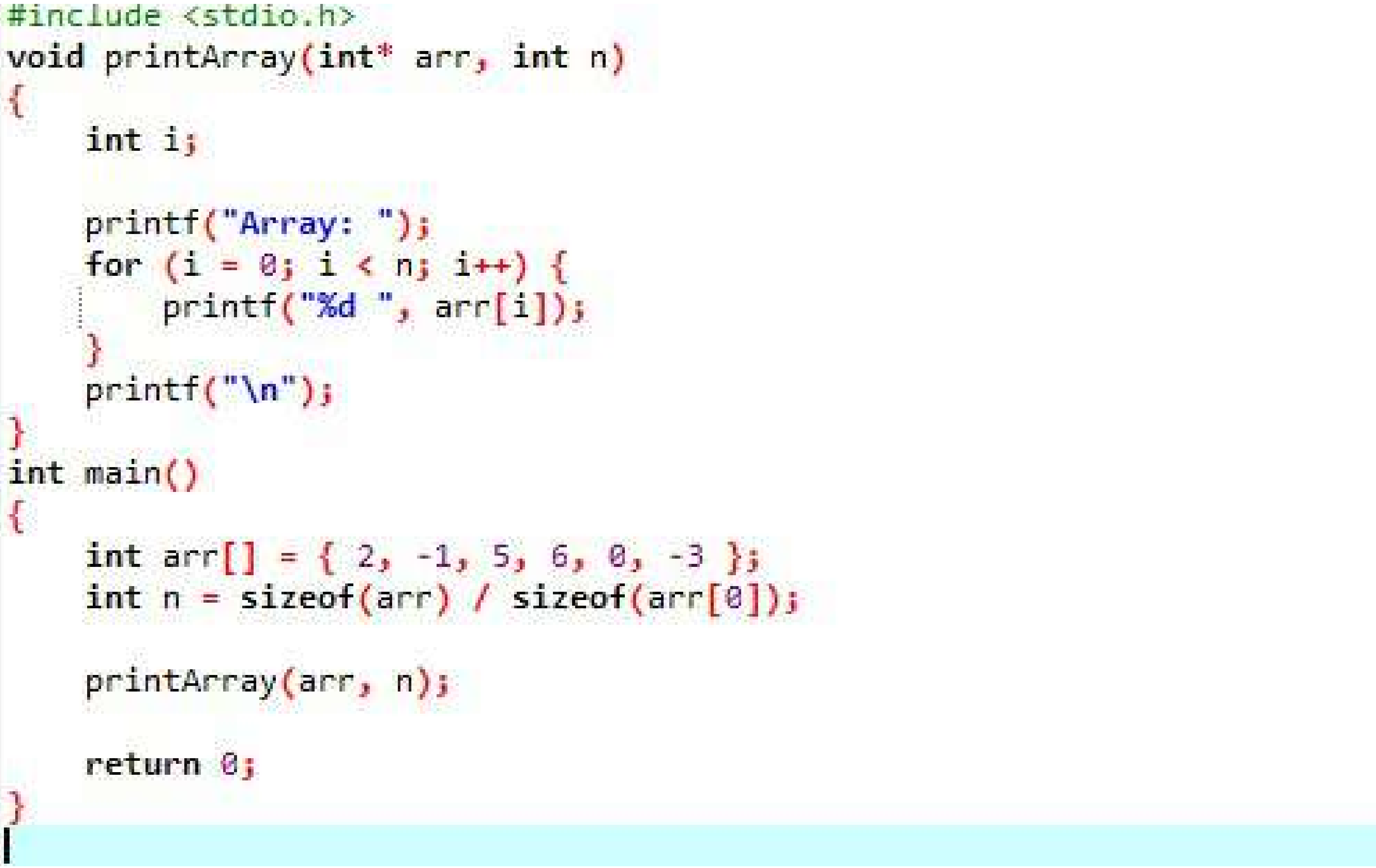
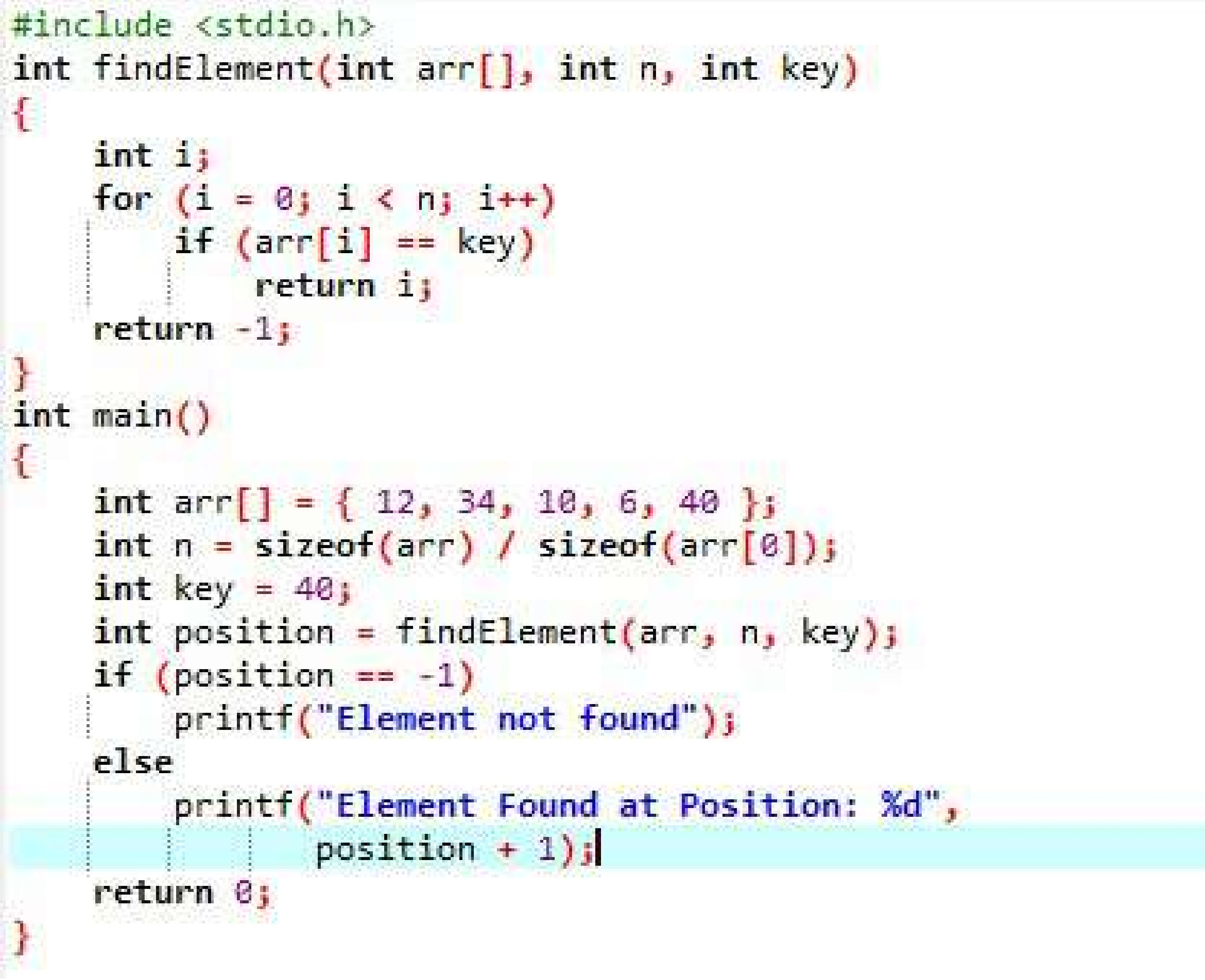
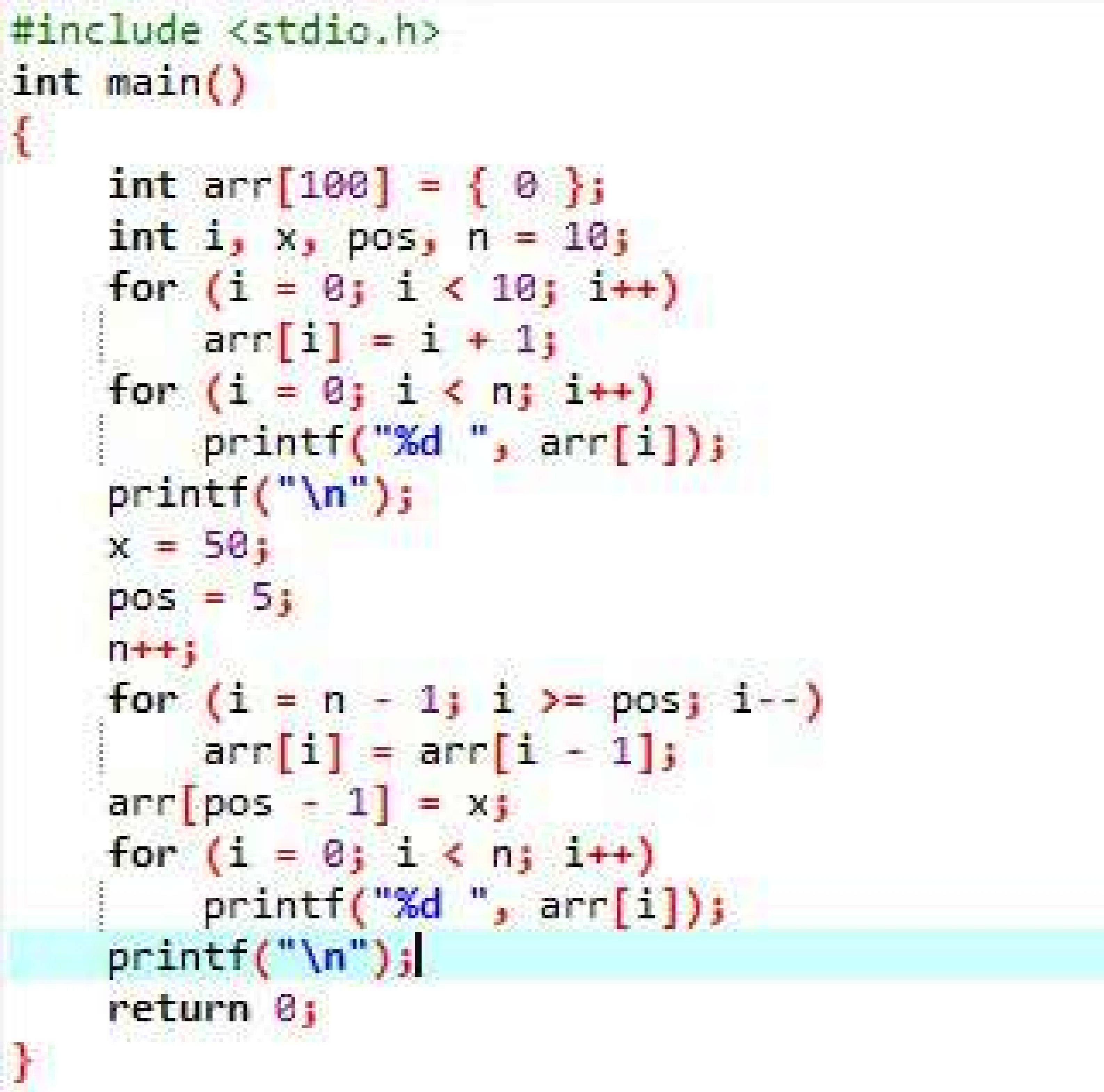


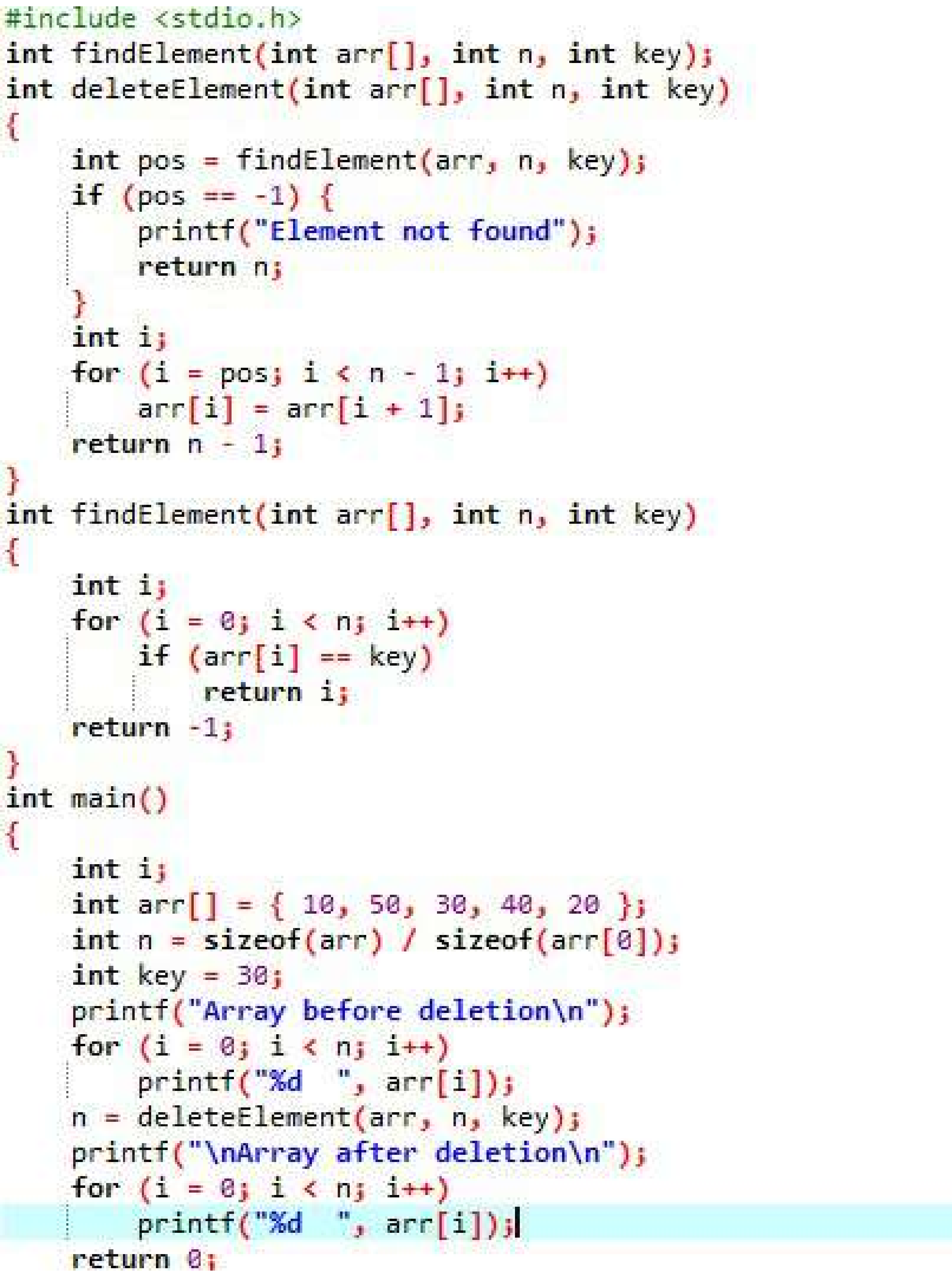
```
#include <stdio.h>
void printArray(int* arr, int n)
{
    int i;

    printf("Array: ");
    for (i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
int main()
{
    int arr[] = { 2, -1, 5, 6, 0, -3 };
    int n = sizeof(arr) / sizeof(arr[0]);

    printArray(arr, n);

    return 0;
}
```



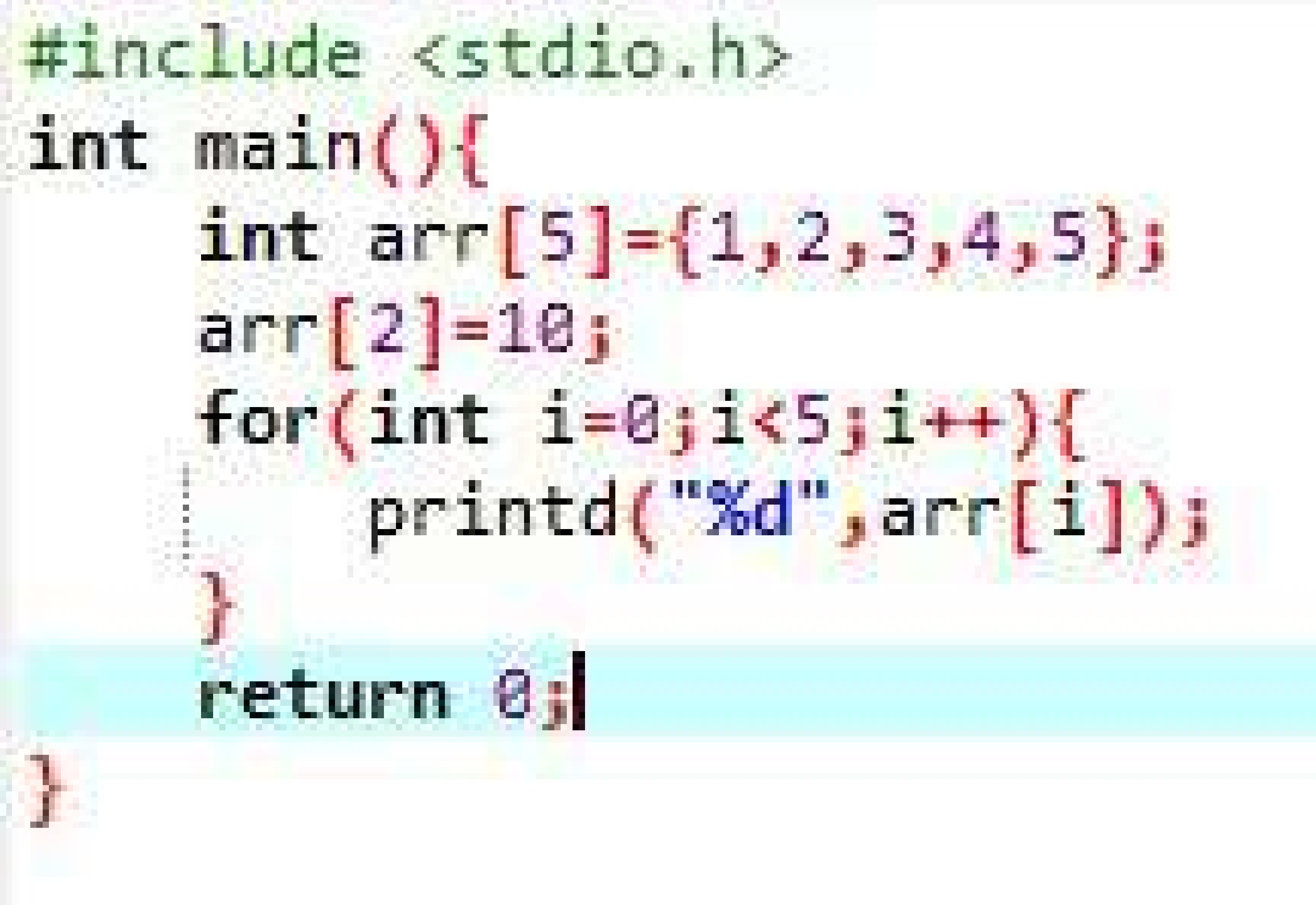
```c
#include <stdio.h>
int findElement(int arr[], int n, int key)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == key)
            return i;
    return -1;
}
int main()
{
    int arr[] = { 12, 34, 10, 6, 40 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 40;
    int position = findElement(arr, n, key);
    if (position == -1)
        printf("Element not found");
    else
        printf("Element Found at Position: %d",
               position + 1);
    return 0;
}
```



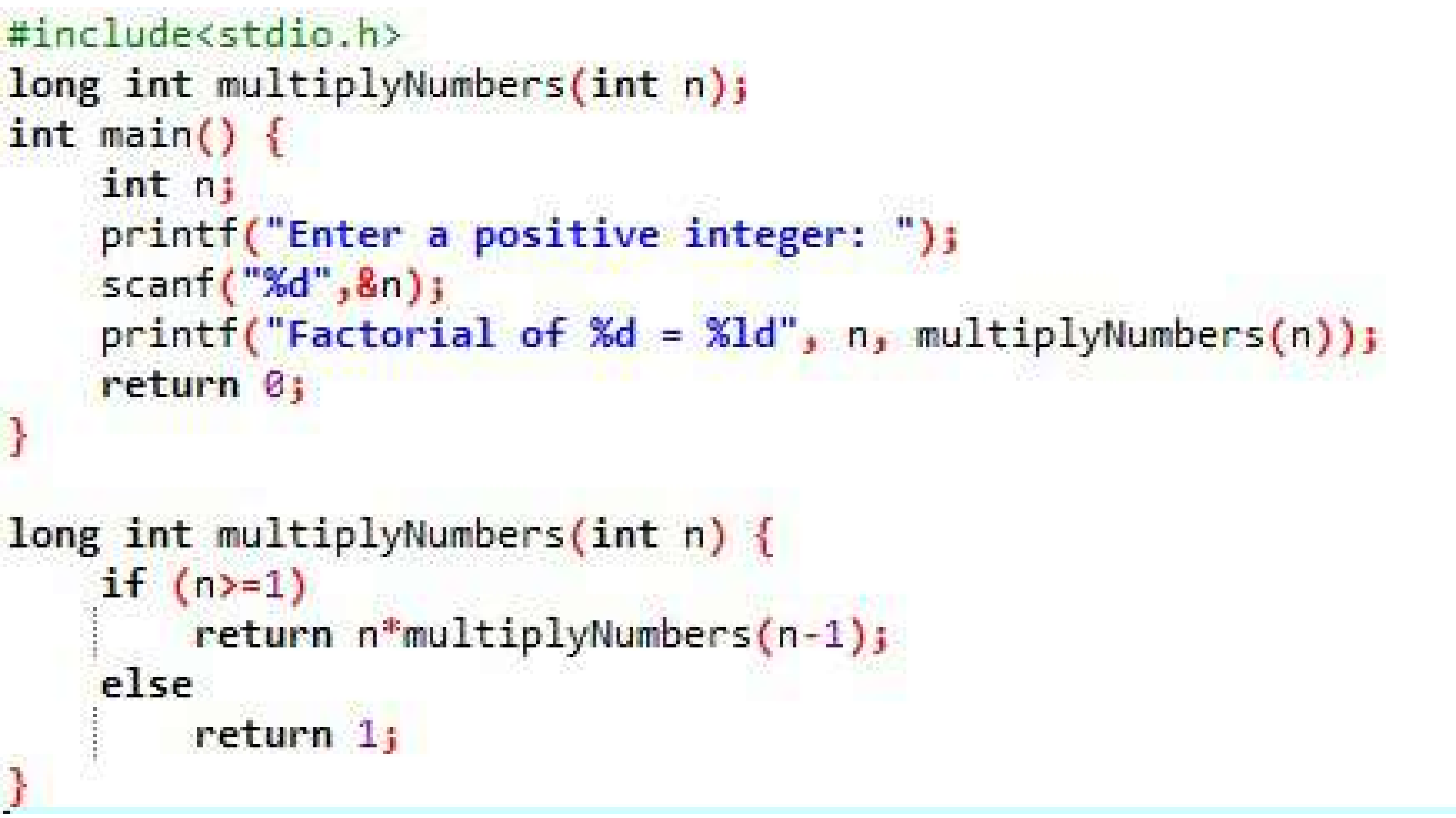
```
#include <stdio.h>
int main()
{
    int arr[100] = { 0 };
    int i, x, pos, n = 10;
    for (i = 0; i < 10; i++)
        arr[i] = i + 1;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    x = 50;
    pos = 5;
    n++;
    for (i = n - 1; i >= pos; i--)
        arr[i] = arr[i - 1];
    arr[pos - 1] = x;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```



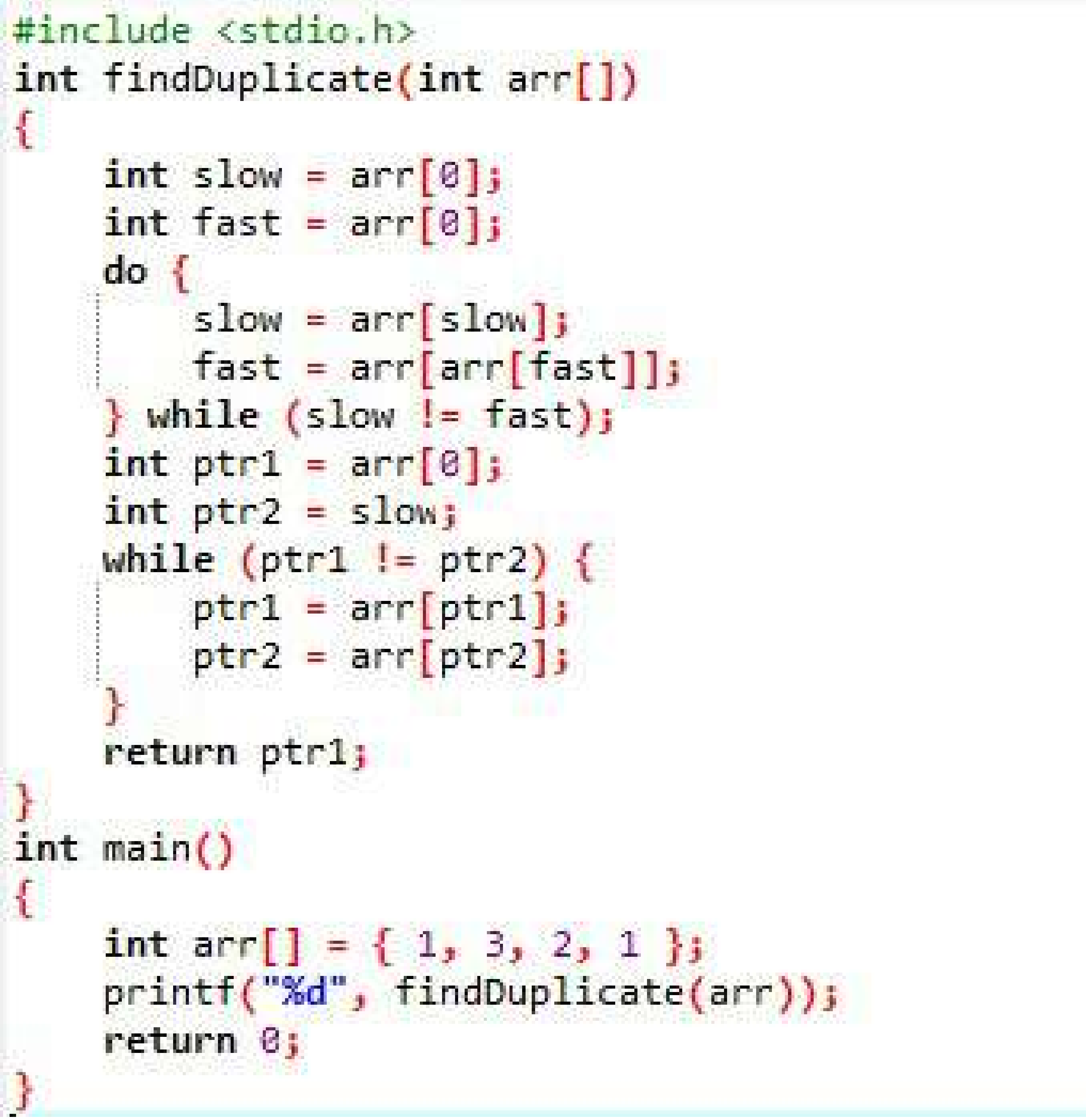
```c
#include <stdio.h>
int findElement(int arr[], int n, int key);
int deleteElement(int arr[], int n, int key)
{
    int pos = findElement(arr, n, key);
    if (pos == -1) {
        printf("Element not found");
        return n;
    }
    int i;
    for (i = pos; i < n - 1; i++)
        arr[i] = arr[i + 1];
    return n - 1;
}
int findElement(int arr[], int n, int key)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == key)
            return i;
    return -1;
}
int main()
{
    int i;
    int arr[] = { 10, 50, 30, 40, 20 };
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 30;
    printf("Array before deletion\n");
    for (i = 0; i < n; i++)
        printf("%d  ", arr[i]);
    n = deleteElement(arr, n, key);
    printf("\nArray after deletion\n");
    for (i = 0; i < n; i++)
        printf("%d  ", arr[i]);
    return 0;
```



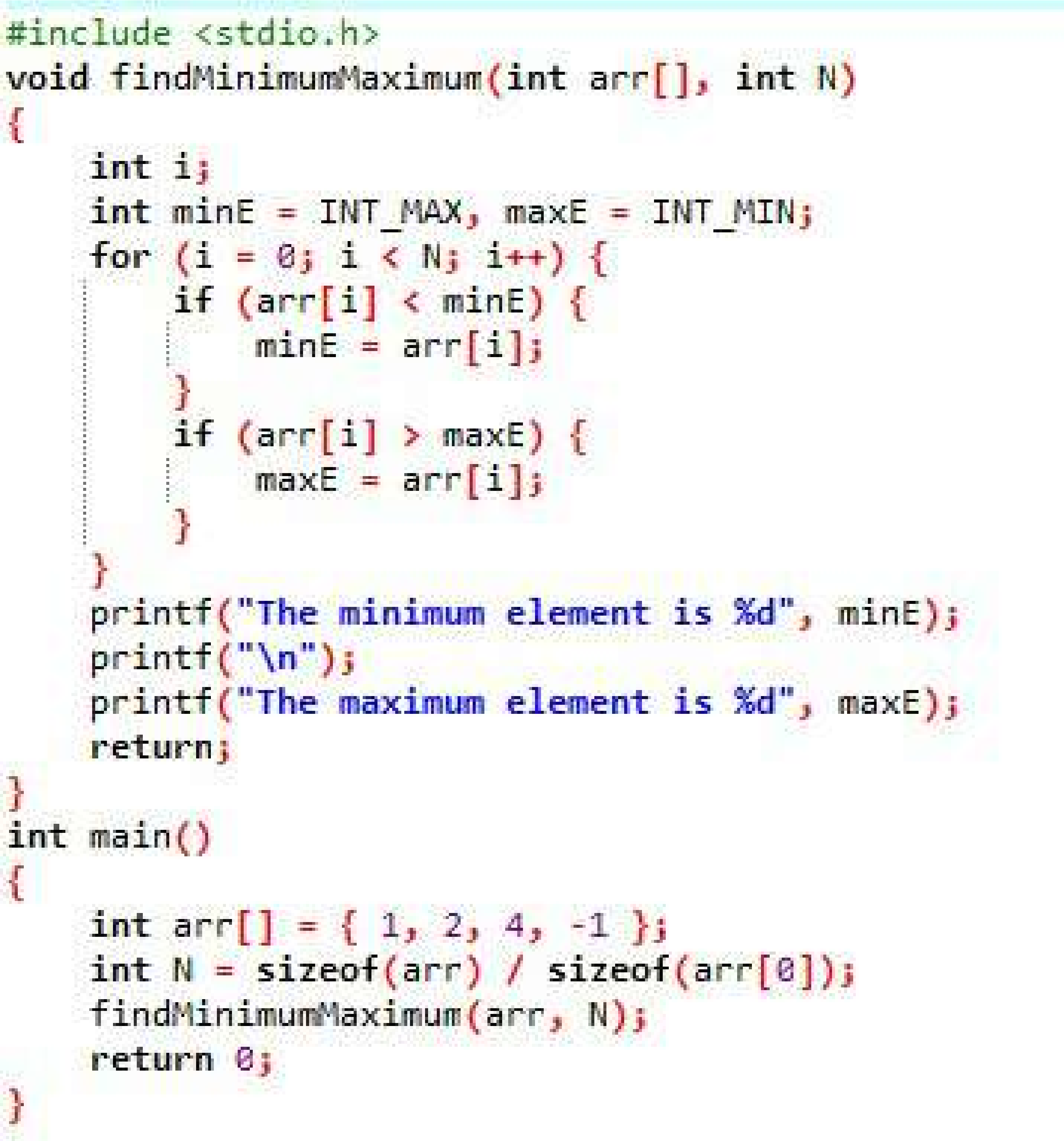
```
#include <stdio.h>
int main(){
    int arr[5]={1,2,3,4,5};
    arr[2]=10;
    for(int i=0;i<5;i++){
        printd("%d",arr[i]);
    }
    return 0;
}
```



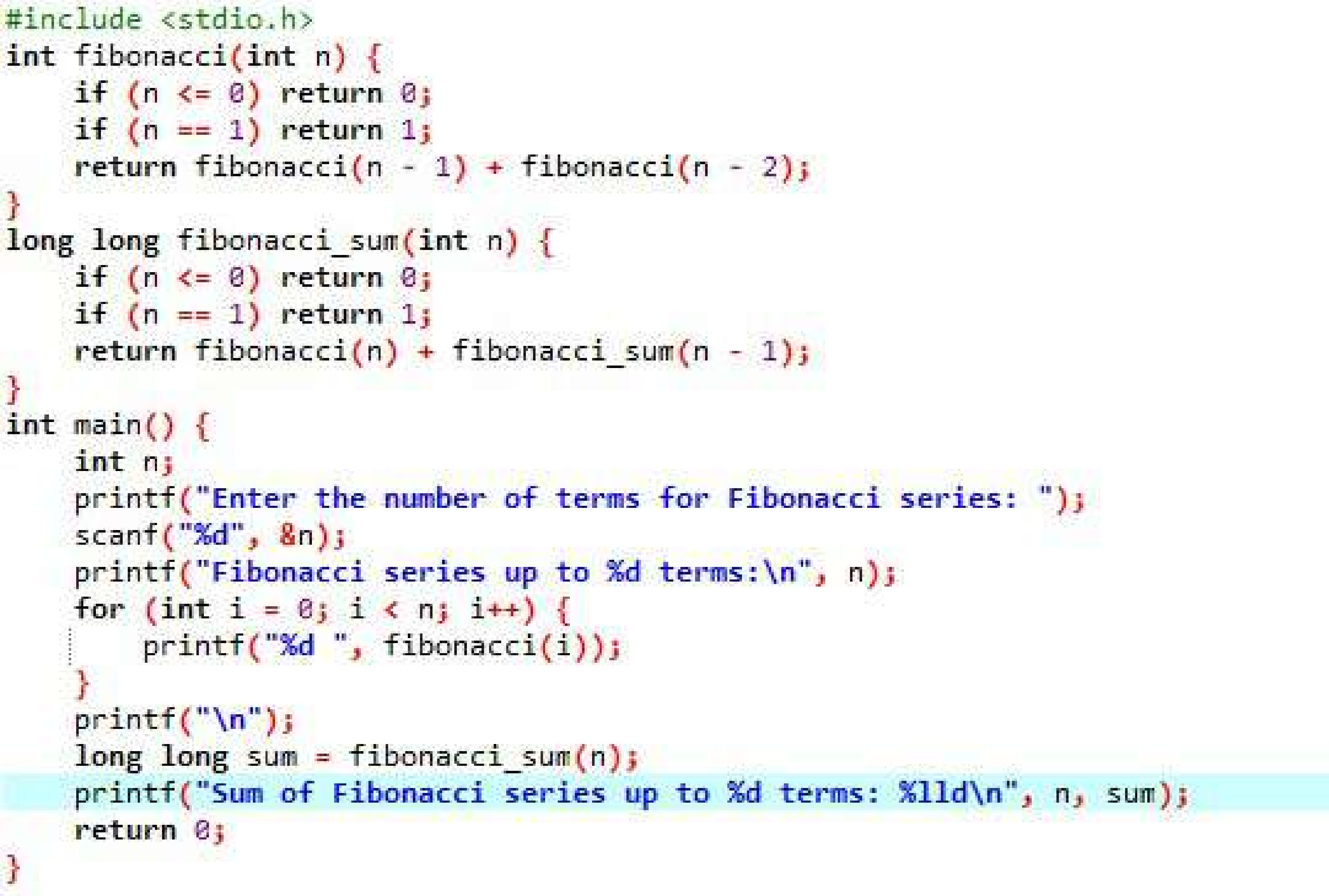
```c
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```



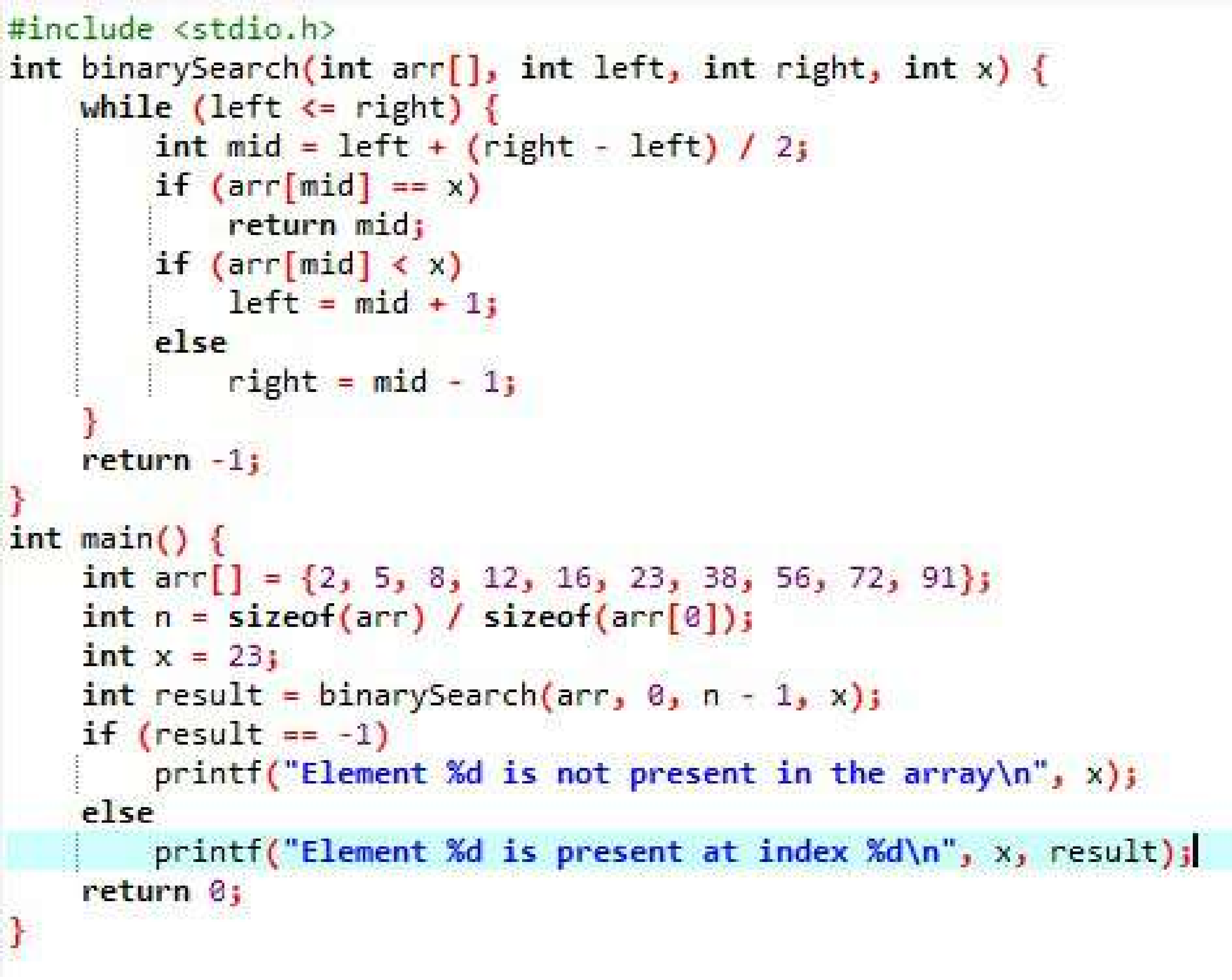
```
#include <stdio.h>
int findDuplicate(int arr[])
{
    int slow = arr[0];
    int fast = arr[0];
    do {
        slow = arr[slow];
        fast = arr[arr[fast]];
    } while (slow != fast);
    int ptr1 = arr[0];
    int ptr2 = slow;
    while (ptr1 != ptr2) {
        ptr1 = arr[ptr1];
        ptr2 = arr[ptr2];
    }
    return ptr1;
}
int main()
{
    int arr[] = { 1, 3, 2, 1 };
    printf("%d", findDuplicate(arr));
    return 0;
}
```



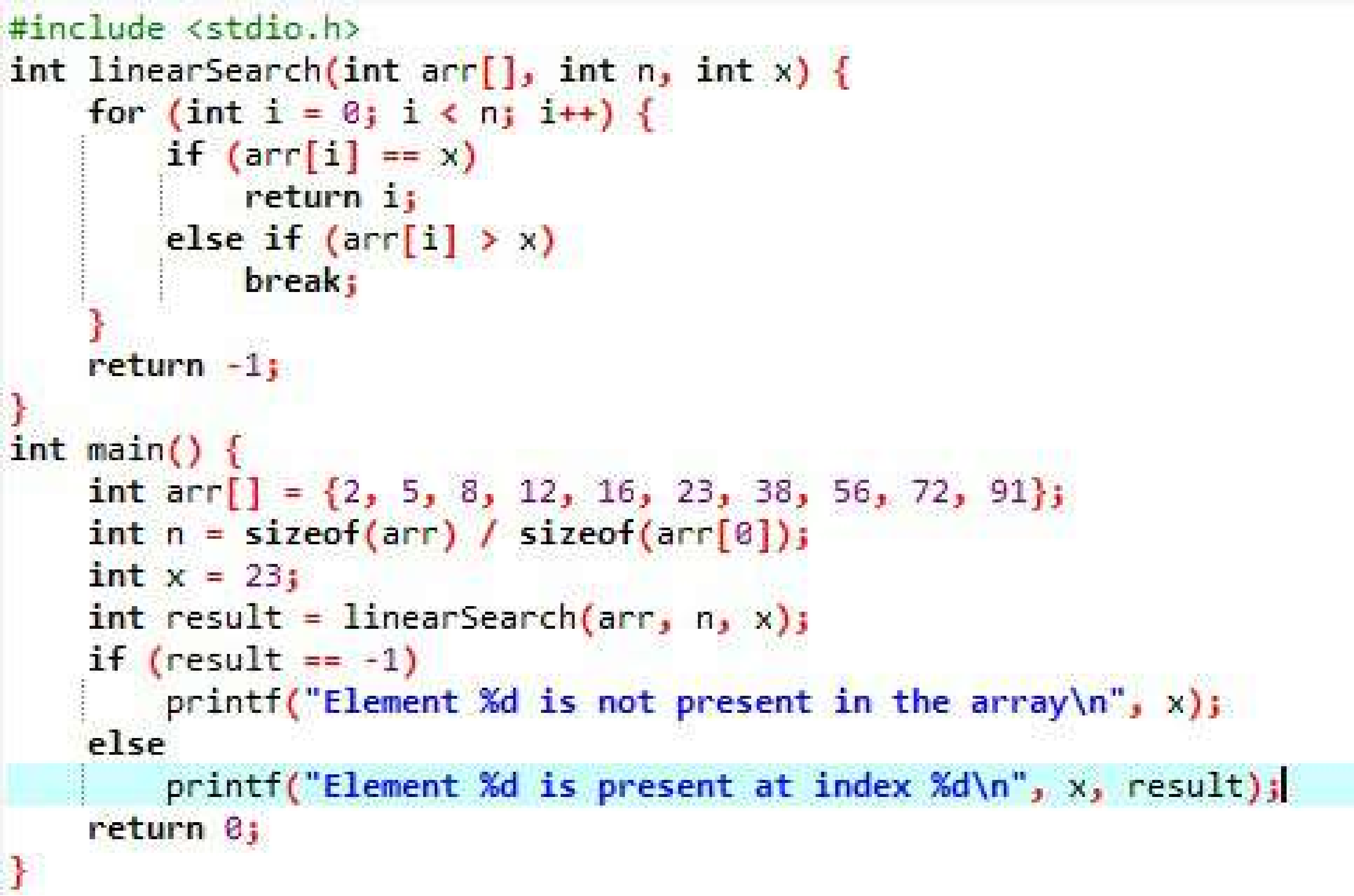
```
#include <stdio.h>
void findMinimumMaximum(int arr[], int N)
{
    int i;
    int minE = INT_MAX, maxE = INT_MIN;
    for (i = 0; i < N; i++) {
        if (arr[i] < minE) {
            minE = arr[i];
        }
        if (arr[i] > maxE) {
            maxE = arr[i];
        }
    }
    printf("The minimum element is %d", minE);
    printf("\n");
    printf("The maximum element is %d", maxE);
    return;
}
int main()
{
    int arr[] = { 1, 2, 4, -1 };
    int N = sizeof(arr) / sizeof(arr[0]);
    findMinimumMaximum(arr, N);
    return 0;
}
```



```c
#include <stdio.h>
int fibonacci(int n) {
    if (n <= 0) return 0;
    if (n == 1) return 1;
    return fibonacci(n - 1) + fibonacci(n - 2);
}
long long fibonacci_sum(int n) {
    if (n <= 0) return 0;
    if (n == 1) return 1;
    return fibonacci(n) + fibonacci_sum(n - 1);
}
int main() {
    int n;
    printf("Enter the number of terms for Fibonacci series: ");
    scanf("%d", &n);
    printf("Fibonacci series up to %d terms:\n", n);
    for (int i = 0; i < n; i++) {
        printf("%d ", fibonacci(i));
    }
    printf("\n");
    long long sum = fibonacci_sum(n);
    printf("Sum of Fibonacci series up to %d terms: %lld\n", n, sum);
    return 0;
}
```



```c
#include <stdio.h>
int binarySearch(int arr[], int left, int right, int x) {
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] < x)
            left = mid + 1;
        else
            right = mid - 1;
    }
    return -1;
}
int main() {
    int arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 23;
    int result = binarySearch(arr, 0, n - 1, x);
    if (result == -1)
        printf("Element %d is not present in the array\n", x);
    else
        printf("Element %d is present at index %d\n", x, result);
    return 0;
}
```



```c
#include <stdio.h>
int linearSearch(int arr[], int n, int x) {
    for (int i = 0; i < n; i++) {
        if (arr[i] == x)
            return i;
        else if (arr[i] > x)
            break;
    }
    return -1;
}
int main() {
    int arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
    int n = sizeof(arr) / sizeof(arr[0]);
    int x = 23;
    int result = linearSearch(arr, n, x);
    if (result == -1)
        printf("Element %d is not present in the array\n", x);
    else
        printf("Element %d is present at index %d\n", x, result);
    return 0;
}
```