

1. Implement Linear Search

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

```
int linear_search(int arr[], int n, int target) {  
    for (int i = 0; i < n; i++) {  
        if (arr[i] == target) {  
            return i;  
        }  
    }  
    return -1;  
}
```

```
int main() {  
    int my_array[] = {5, 2, 9, 1, 7, 6};  
    int array_size = sizeof(my_array) / sizeof(my_array[0]);  
    int target_element = 7;  
  
    int result = linear_search(my_array, array_size, target_element);  
  
    if (result != -1) {  
        printf("Element %d found at index %d.\n", target_element, result);  
    } else {  
        printf("Element %d not found in the array.\n", target_element);  
    }  
  
    return 0;  
}
```

Output

```
/tmp/h5KfhmElC3.o  
Element 7 found at index 4.
```

2. Implement Binary Search

```
#include <stdio.h>

int binary_search(int arr[], int n, int target) {
    int low = 0, high = n - 1;

    while (low <= high) {
        int mid = (low + high) / 2;

        if (arr[mid] == target) {
            return mid;
        } else if (arr[mid] < target) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }

    return -1;
}

int main() {
    int sorted_array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int array_size = sizeof(sorted_array) / sizeof(sorted_array[0]);
    int target_element = 7;

    int result = binary_search(sorted_array, array_size, target_element);

    if (result != -1) {
        printf("Element %d found at index %d\n", target_element, result);
    } else {
        printf("Element %d not found in the array\n", target_element);
    }
}
```

```
}  
  
    return 0;  
}
```

Output

```
/tmp/lq1f6l9V2h.o  
Element 7 found at index 6
```