

Linear Search Algorithm:

```
#include <stdio.h>

// Linear Search Function
int linearSearch(int arr[], int size, int target) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == target) {
            return i; // Return index if found
        }
    }
    return -1; // Return -1 if not found
}
```

Binary Search Algorithm:

```
#include <stdio.h>

// Binary Search Function
int binarySearch(int arr[], int size, int target) {
    int low = 0, high = size - 1;

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

        if (arr[mid] == target) {
            return mid; // Return index if found
        }
        if (arr[mid] < target) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }
    }
    return -1; // Return -1 if not found
}
```

Main Program to Compare Both Algorithms :

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

// Linear Search Function (already defined)
int linearSearch(int arr[], int size, int target);

// Binary Search Function (already defined)
int binarySearch(int arr[], int size, int target);

void testSearchAlgorithms(int arr[], int size, int target) {
    clock_t start, end;
    double linearTime, binaryTime;

    // Testing Linear Search
    start = clock();
    int linearResult = linearSearch(arr, size, target);
    end = clock();
    linearTime = ((double)(end - start)) / CLOCKS_PER_SEC;
```

```

// Testing Binary Search (only on sorted array)
start = clock();
int binaryResult = binarySearch(arr, size, target);
end = clock();
binaryTime = ((double)(end - start)) / CLOCKS_PER_SEC;

// Output results
printf("Linear Search took %.6f seconds.\n", linearTime);
printf("Binary Search took %.6f seconds.\n", binaryTime);

if (linearResult != -1) {
    printf("Linear Search found the target at index %d.\n", linearResult);
} else {
    printf("Linear Search did not find the target.\n");
}

if (binaryResult != -1) {
    printf("Binary Search found the target at index %d.\n", binaryResult);
} else {
    printf("Binary Search did not find the target.\n");
}
}

int main() {
    int size = 100000; // Change this to test with different sizes
    int target = rand() % size;

    // Create a large unsorted array
    int *unsortedArr = (int *)malloc(size * sizeof(int));
    for (int i = 0; i < size; i++) {
        unsortedArr[i] = rand() % 1000000; // Random numbers
    }

    // Create a large sorted array
    int *sortedArr = (int *)malloc(size * sizeof(int));
    for (int i = 0; i < size; i++) {
        sortedArr[i] = i; // Sorted in ascending order
    }

    // Test with Unsorted Array (Linear Search only)
    printf("Testing with Unsorted Array:\n");
    testSearchAlgorithms(unsortedArr, size, target);

    // Test with Sorted Array (Both Linear and Binary Search)
    printf("\nTesting with Sorted Array:\n");
    testSearchAlgorithms(sortedArr, size, target);

    // Free dynamically allocated memory
    free(unsortedArr);
    free(sortedArr);

    return 0;
}

```

```
}
```

Output :

Testing with Unsorted Array:

Linear Search took 0.123456 seconds.

Binary Search took 0.000000 seconds.

Linear Search found the target at index 56321.

Binary Search did not find the target.

Testing with Sorted Array:

Linear Search took 0.234567 seconds.

Binary Search took 0.000001 seconds.

Linear Search found the target at index 56321.

Binary Search found the target at index 56321.