Description:

C program that creates a sorted array, uses binary search to find a target in O(log n), and reports both the result (index or not found) and the CPU time measured with clock().

Source Code:

```
#include <stdio.h>

#include <time.h>

int binarySearch(int arr[], int n, int key) {

        int low = 0;

        int high = n - 1;


        while (low <= high) {

        int mid = (high + low) / 2;


        if (arr[mid] == key) {

        return mid;

        } else if (arr[mid] < key) {

        low = mid + 1;

        } else {

        high = mid - 1;

        }

        }


        return -1;

}

int main() {
```

```
        int n = 100000;
        int arr[n];

for (int i = 0; i < n; i++) {
        arr[i] = i;
        }

        int key = 99999;

        clock_t start, end;
        double cpu_time_used;

        start = clock();

        int result = binarySearch(arr, n, key);

        end = clock();

        cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;

        if (result != -1) {
        printf("Element found at index %d\n", result);
        } else {
        printf("Element not found\n");
        }
        printf("Execution time: %.6f seconds\n", cpu_time_used);
```

```
    return 0;
}
```

Output:

```
Element found at index 99999
Execution time: 0.000002 seconds
```