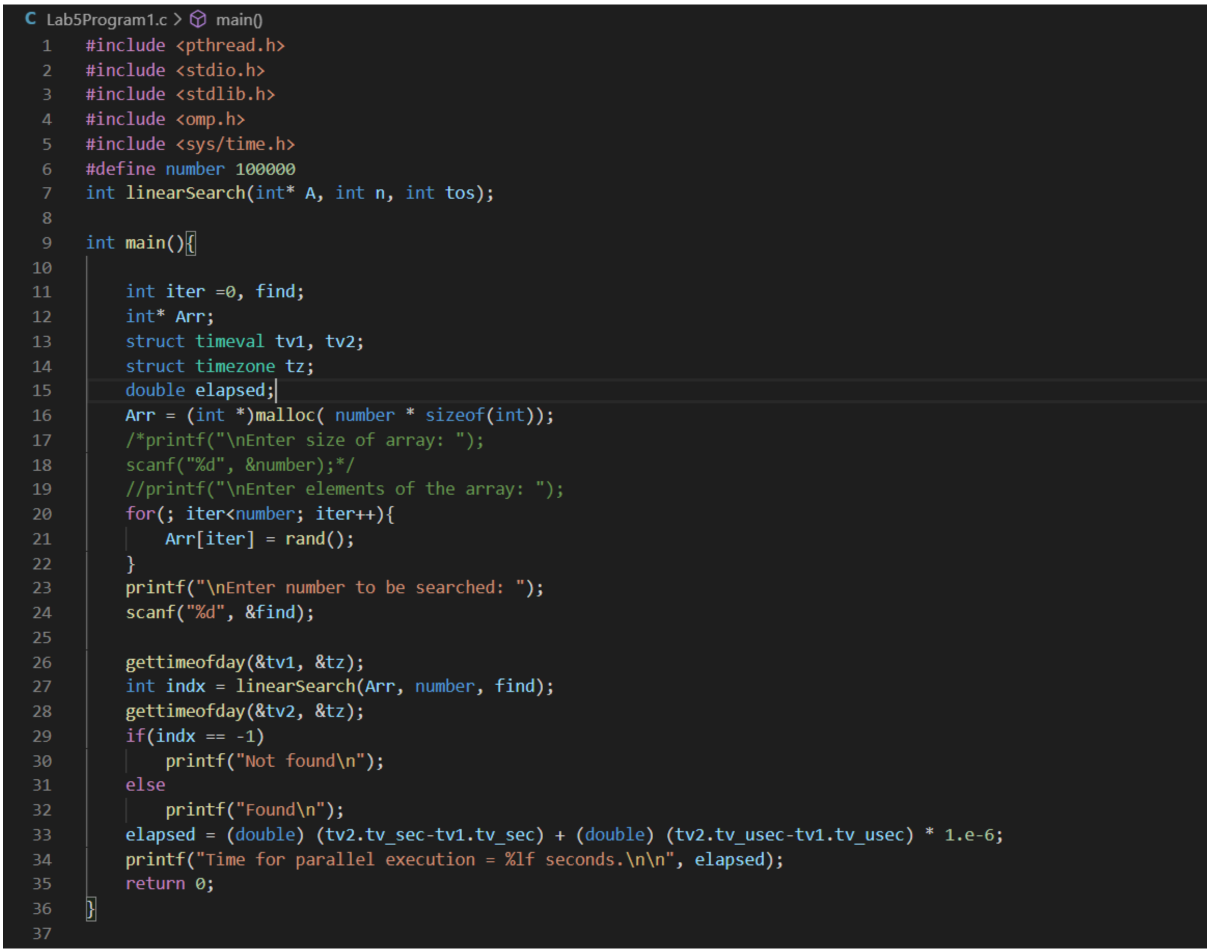
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL
DEPARTMENT OF INFORMATION TECHNOLOGY
IT 301 Parallel Computing LAB 5
9th September 2020
Faculty: Dr. Geetha V and Mrs. Thanmayee

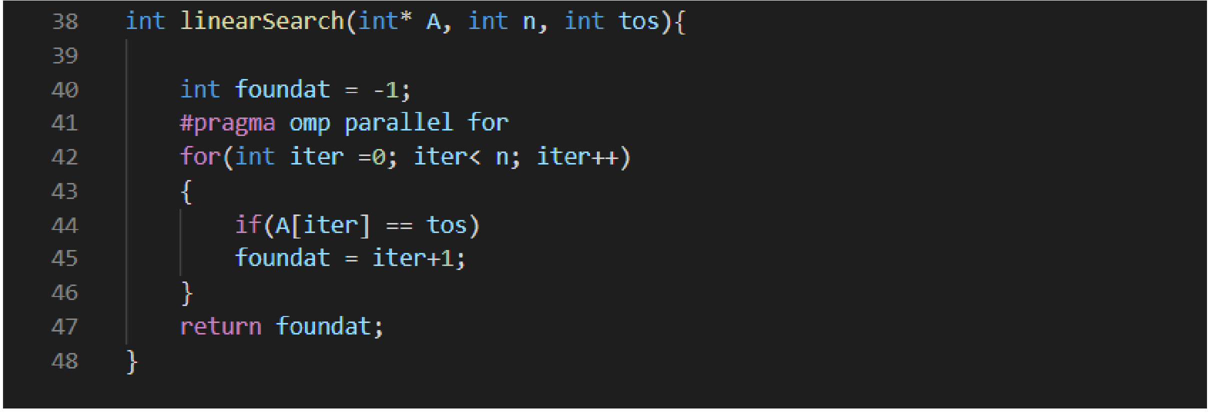
Name: Chinmayi C. Ramakrishna

Roll No.: 181IT113

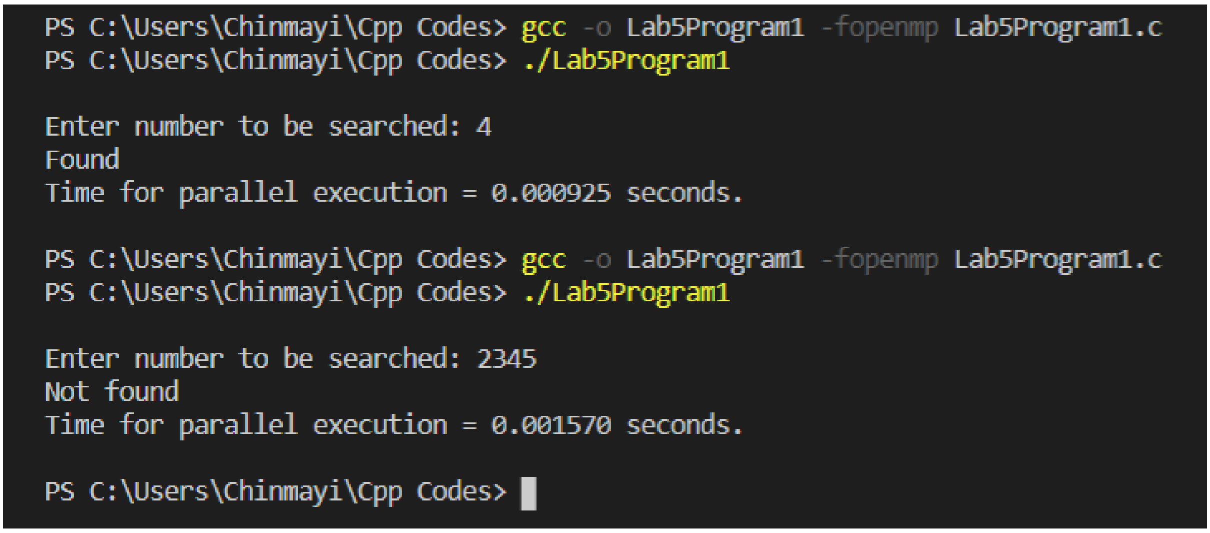
Develop a parallel program to find a given element in an unsorted array (a large number of elements starting from 10K can range to 1 lakh and above, based on the memory) using Linear Search. Compare the execution time with the Sequential Linear Search program. Also compare it with the sequential Binary Search program.



```c
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>
#define number 100000
int linearSearch(int* A, int n, int tos);

int main(){

    int iter =0, find;
    int* Arr;
    struct timeval tv1, tv2;
    struct timezone tz;
    double elapsed;
    Arr = (int *)malloc( number * sizeof(int));
    /*printf("\nEnter size of array: ");
    scanf("%d", &number);*/
    //printf("\nEnter elements of the array: ");
    for(; iter<number; iter++){
        Arr[iter] = rand();
    }
    printf("\nEnter number to be searched: ");
    scanf("%d", &find);

    gettimeofday(&tv1, &tz);
    int indx = linearSearch(Arr, number, find);
    gettimeofday(&tv2, &tz);
    if(indx == -1)
        printf("Not found\n");
    else
        printf("Found\n");
    elapsed = (double) (tv2.tv_sec-tv1.tv_sec) + (double) (tv2.tv_usec-tv1.tv_usec) * 1.e-6;
    printf("Time for parallel execution = %lf seconds.\n\n", elapsed);
    return 0;
}

```

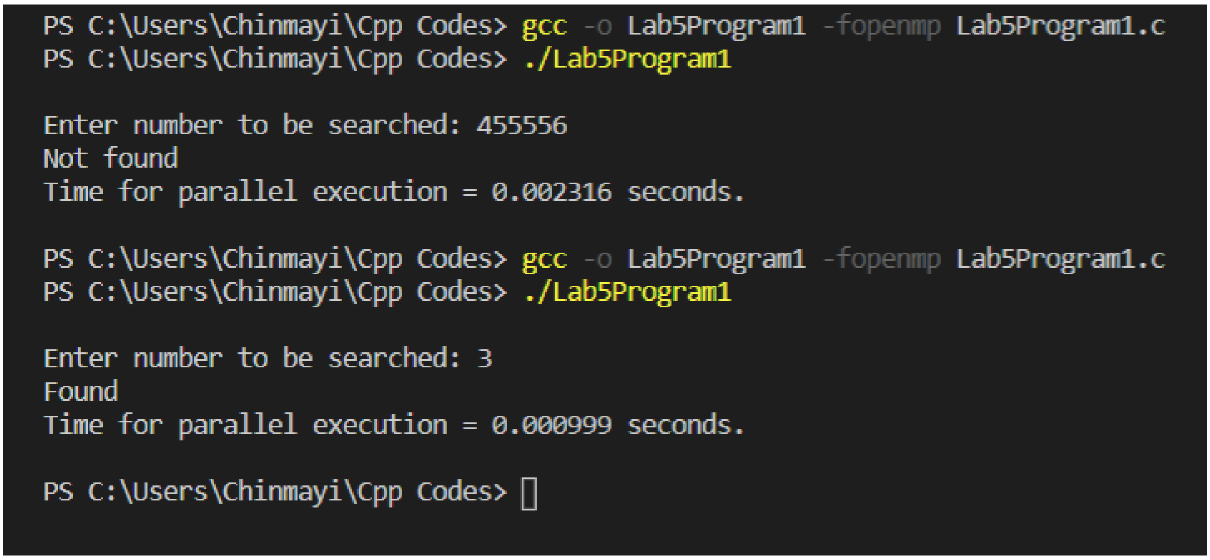


```c
int linearSearch(int* A, int n, int tos){

    int foundat = -1;
    #pragma omp parallel for
    for(int iter =0; iter< n; iter++)
    {
        if(A[iter] == tos)
        foundat = iter+1;
    }
    return foundat;
}
```



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 4
Found
Time for parallel execution = 0.000925 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 2345
Not found
Time for parallel execution = 0.001570 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

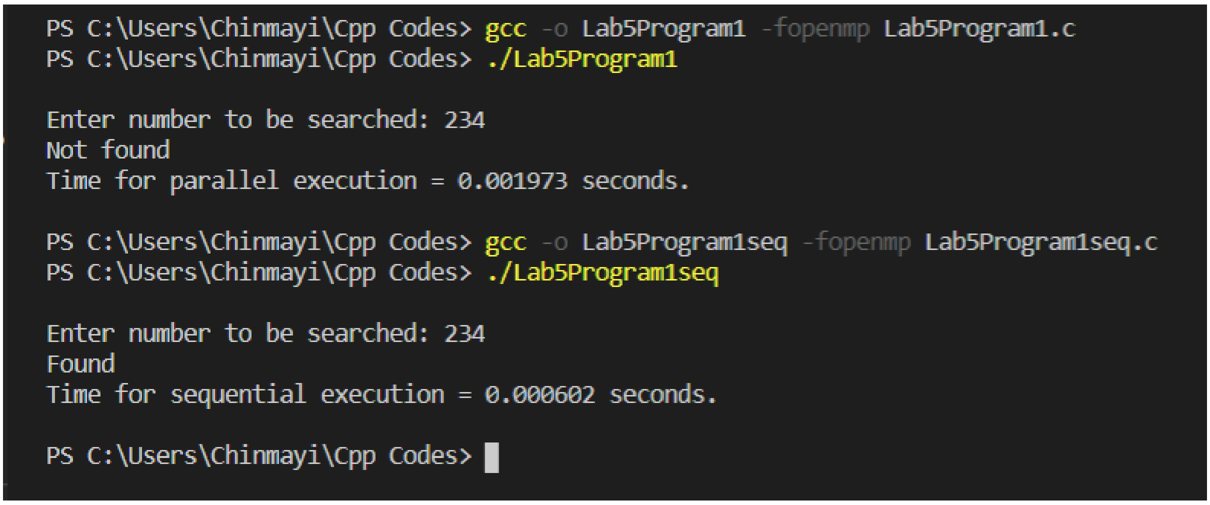
Array size = 10K.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 455556
Not found
Time for parallel execution = 0.002316 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 3
Found
Time for parallel execution = 0.000999 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

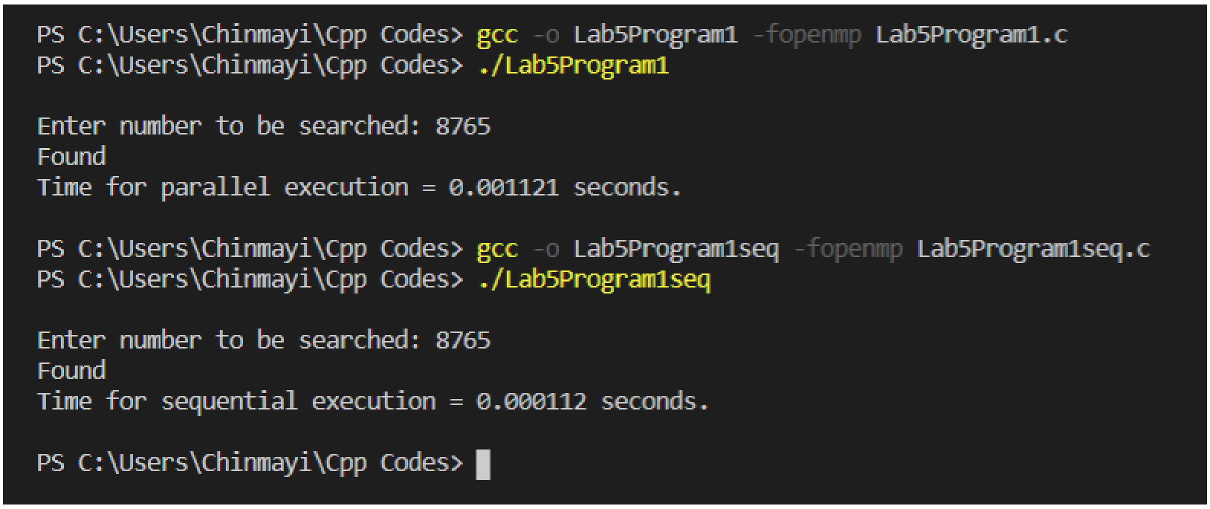
Array size = 1 Lakh.

Comparing Parallel Linear Search with sequential Linear Search.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 234
Not found
Time for parallel execution = 0.001973 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1seq -fopenmp Lab5Program1seq.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1seq

Enter number to be searched: 234
Found
Time for sequential execution = 0.000602 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

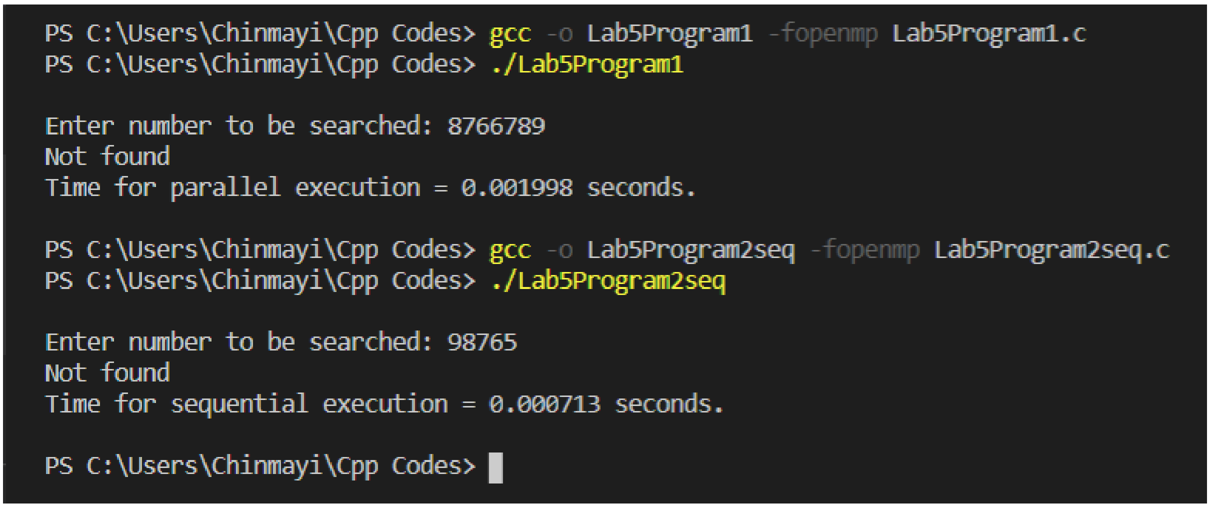
Array size = 10K.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 8765
Found
Time for parallel execution = 0.001121 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1seq -fopenmp Lab5Program1seq.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1seq

Enter number to be searched: 8765
Found
Time for sequential execution = 0.000112 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

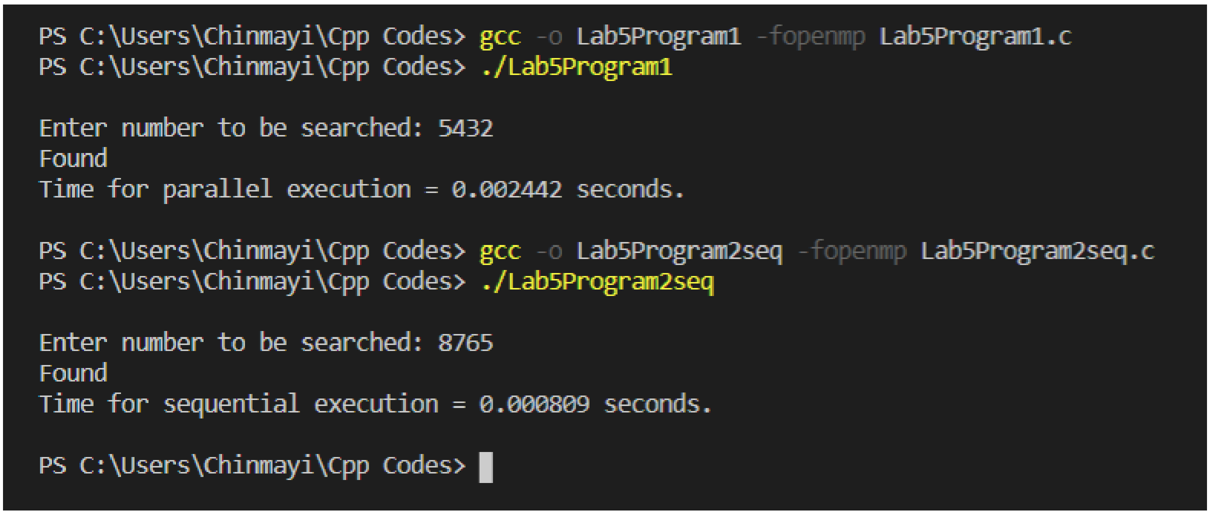
Array size = 1Lakh.

Comparing Parallel Linear Search with Sequential Binary Search.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 8766789
Not found
Time for parallel execution = 0.001998 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2seq -fopenmp Lab5Program2seq.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2seq

Enter number to be searched: 98765
Not found
Time for sequential execution = 0.000713 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

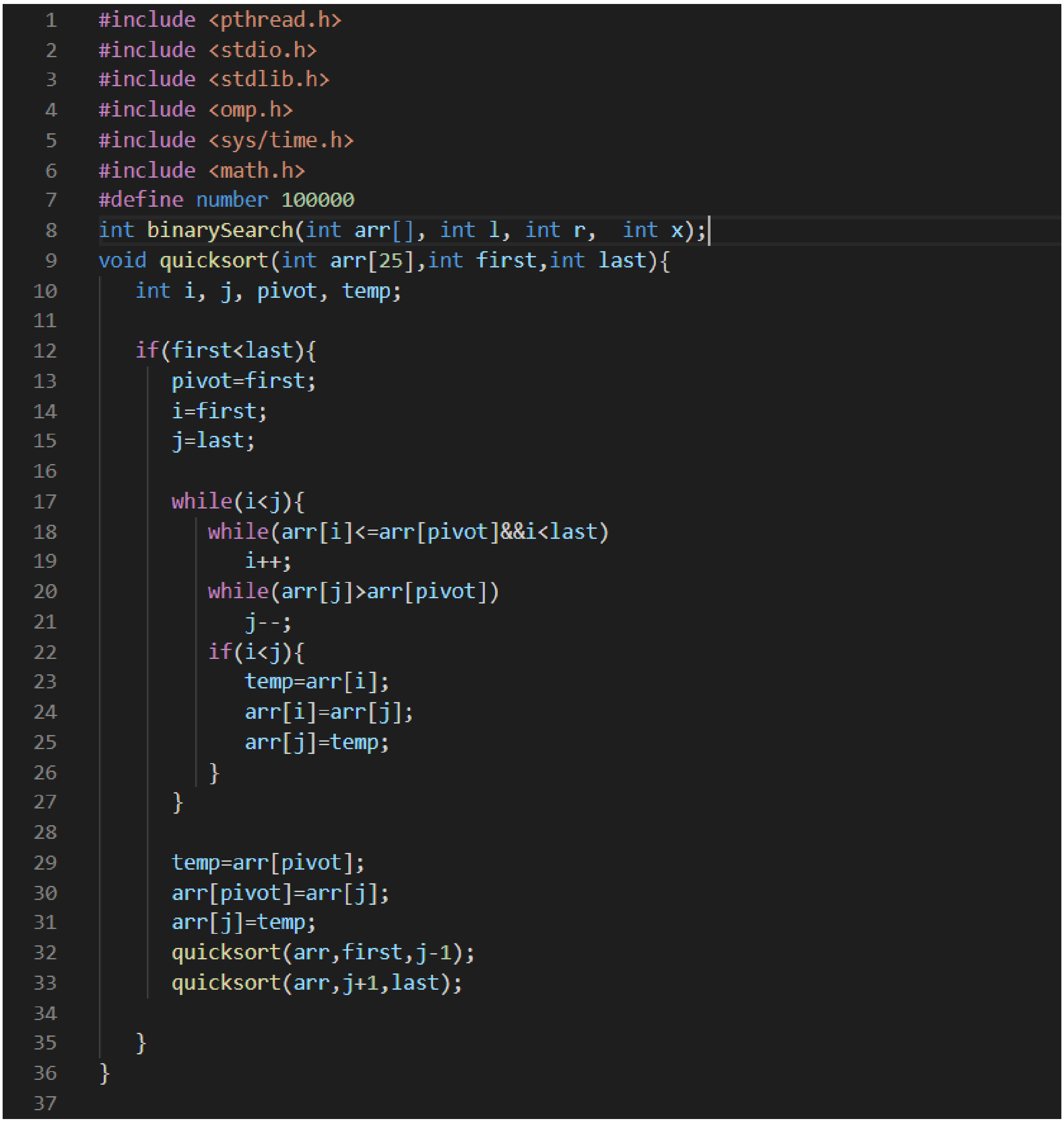
Array size = 10K.



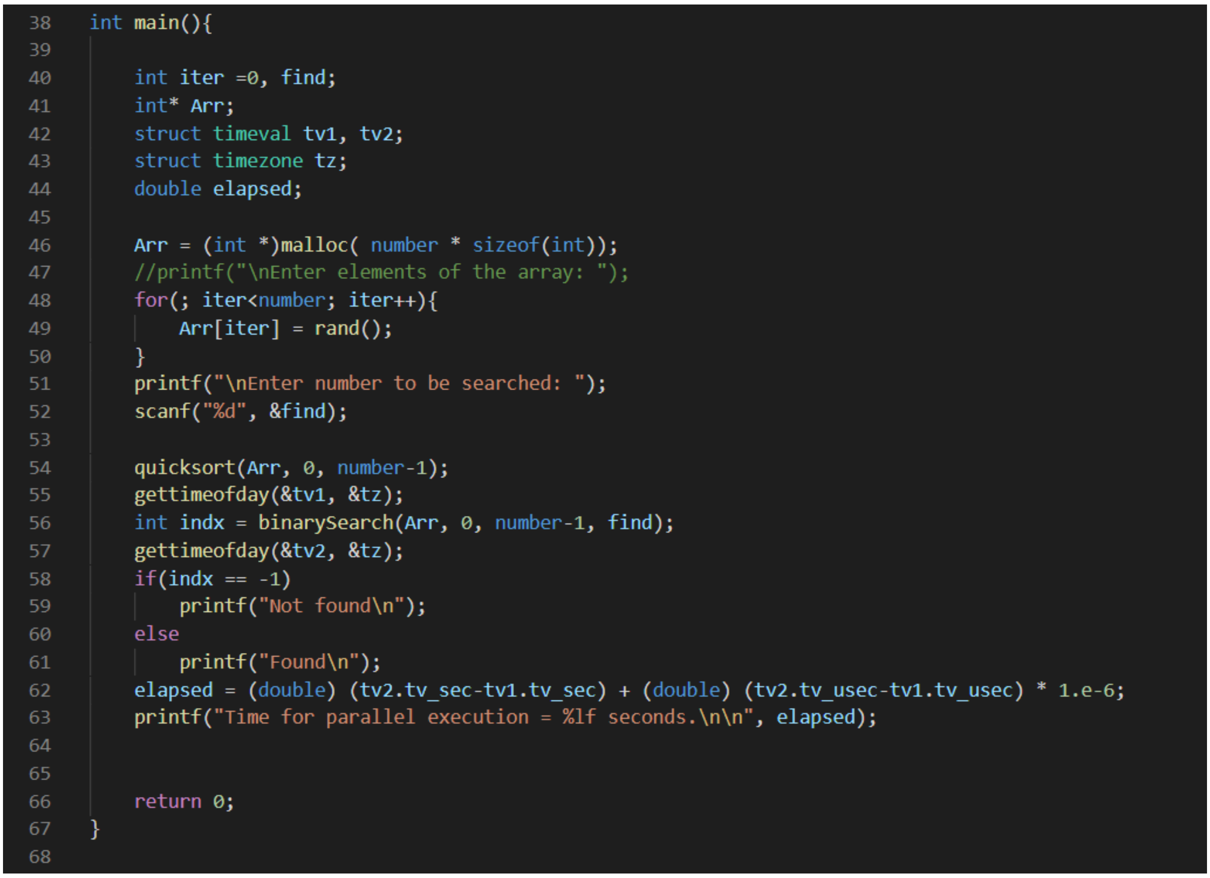
```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program1 -fopenmp Lab5Program1.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program1

Enter number to be searched: 5432
Found
Time for parallel execution = 0.002442 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2seq -fopenmp Lab5Program2seq.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2seq

Enter number to be searched: 8765
Found
Time for sequential execution = 0.000809 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

Array size = 1Lakh.

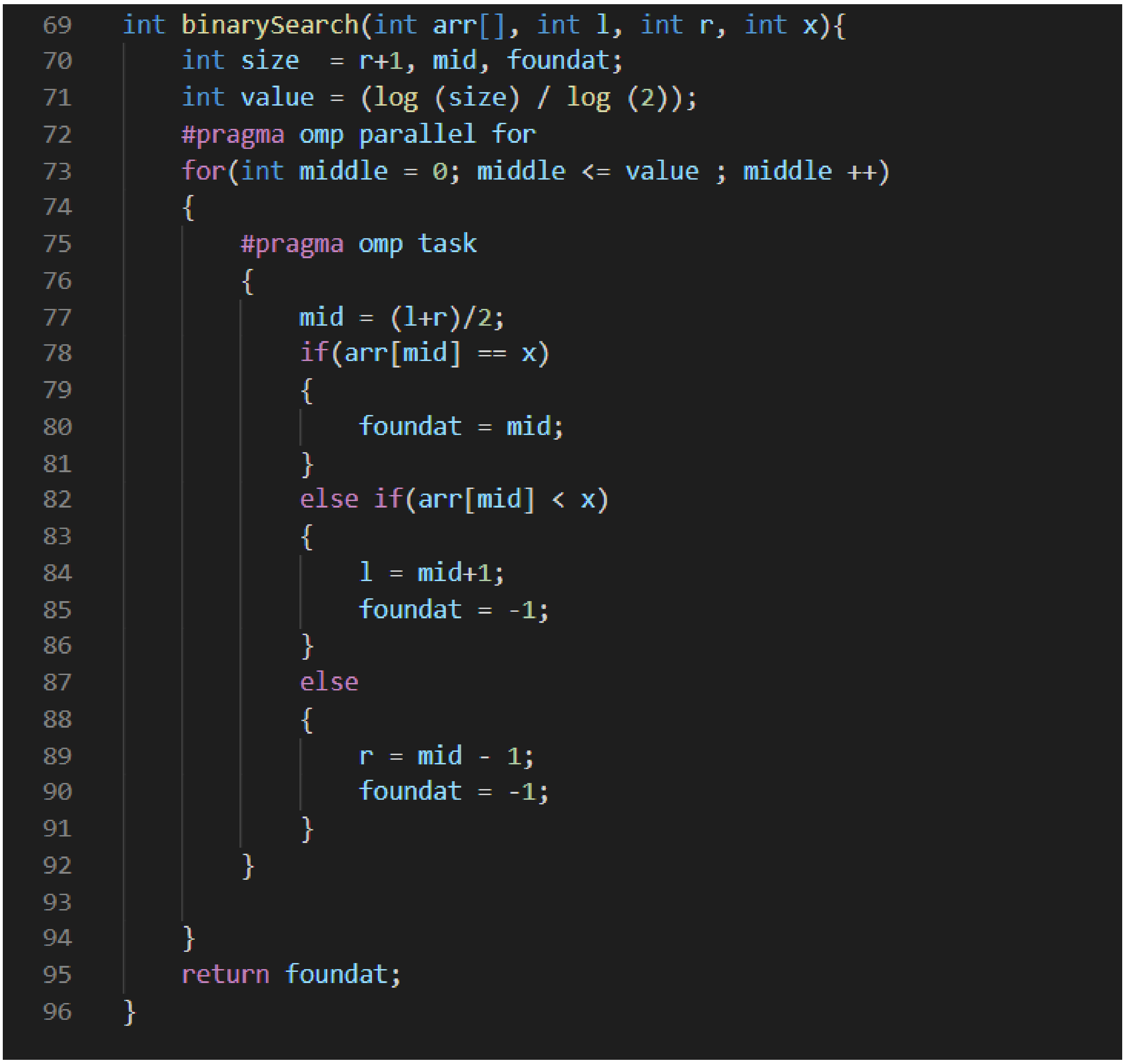
Develop a parallel program to find a given element in an unsorted array using Binary Search. Take a large number of elements up to the maximum possible size. Note: Make use of openmp task directive. Also compare the execution time with the sequential version of Binary Search.



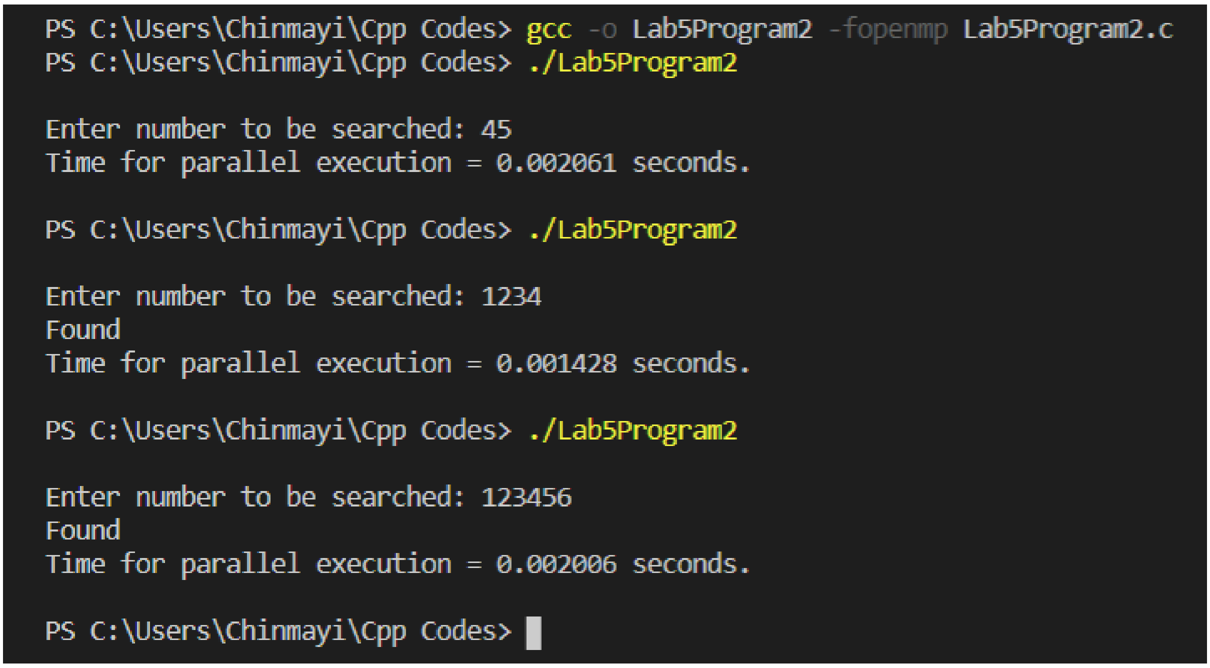
```
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <sys/time.h>
#include <math.h>
#define number 100000
int binarySearch(int arr[], int l, int r,  int x);
void quicksort(int arr[25],int first,int last){
    int i, j, pivot, temp;

    if(first<last){
        pivot=first;
        i=first;
        j=last;

        while(i<j){
            while(arr[i]<=arr[pivot]&&i<last)
                i++;
            while(arr[j]>arr[pivot])
                j--;
            if(i<j){
                temp=arr[i];
                arr[i]=arr[j];
                arr[j]=temp;
            }
        }

        temp=arr[pivot];
        arr[pivot]=arr[j];
        arr[j]=temp;
        quicksort(arr,first,j-1);
        quicksort(arr,j+1,last);

    }
}

```



```
int main(){

    int iter =0, find;
    int* Arr;
    struct timeval tv1, tv2;
    struct timezone tz;
    double elapsed;

    Arr = (int *)malloc( number * sizeof(int));
    //printf("\nEnter elements of the array: ");
    for(; iter<number; iter++){
        Arr[iter] = rand();
    }
    printf("\nEnter number to be searched: ");
    scanf("%d", &find);

    quicksort(Arr, 0, number-1);
    gettimeofday(&tv1, &tz);
    int indx = binarySearch(Arr, 0, number-1, find);
    gettimeofday(&tv2, &tz);
    if(indx == -1)
        printf("Not found\n");
    else
        printf("Found\n");
    elapsed = (double) (tv2.tv_sec-tv1.tv_sec) + (double) (tv2.tv_usec-tv1.tv_usec) * 1.e-6;
    printf("Time for parallel execution = %lf seconds.\n\n", elapsed);


    return 0;
}

```

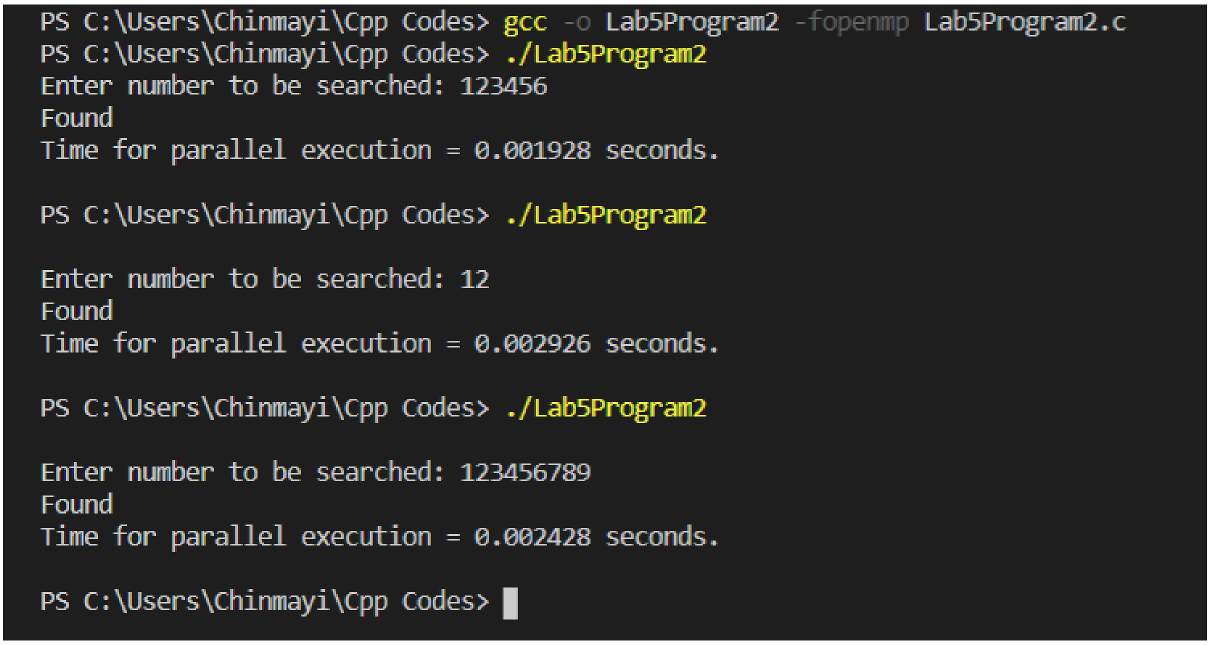


```
int binarySearch(int arr[], int l, int r, int x){
    int size  = r+1, mid, foundat;
    int value = (log (size) / log (2));
    #pragma omp parallel for
    for(int middle = 0; middle <= value ; middle ++)
    {
        #pragma omp task
        {
            mid = (l+r)/2;
            if(arr[mid] == x)
            {
                foundat = mid;
            }
            else if(arr[mid] < x)
            {
                l = mid+1;
                foundat = -1;
            }
            else
            {
                r = mid - 1;
                foundat = -1;
            }
        }

    }
    return foundat;
}
```



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2 -fopenmp Lab5Program2.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 45
Time for parallel execution = 0.002061 seconds.

PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 1234
Found
Time for parallel execution = 0.001428 seconds.

PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 123456
Found
Time for parallel execution = 0.002006 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

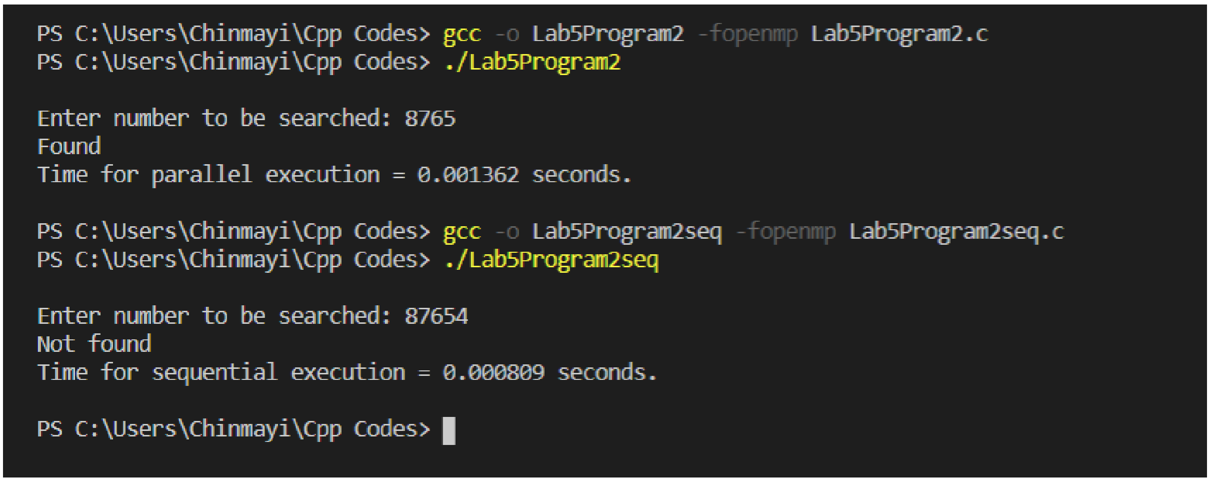
Array size = 10K.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2 -fopenmp Lab5Program2.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2
Enter number to be searched: 123456
Found
Time for parallel execution = 0.001928 seconds.

PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 12
Found
Time for parallel execution = 0.002926 seconds.

PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 123456789
Found
Time for parallel execution = 0.002428 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

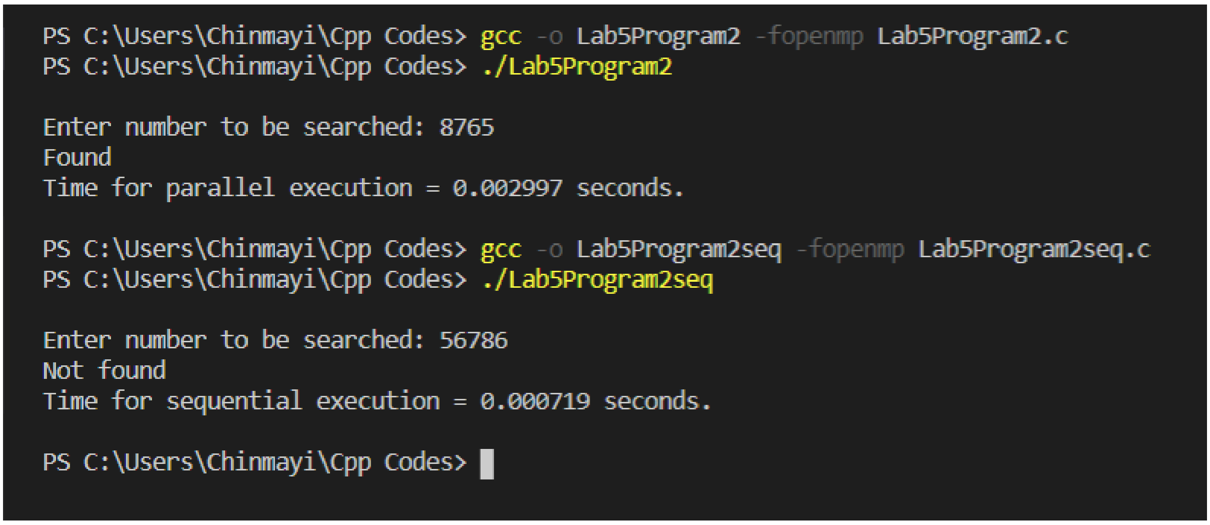
Array size = 1 Lakh.

Comparing Parallel Binary Search with Sequential Binary Search.



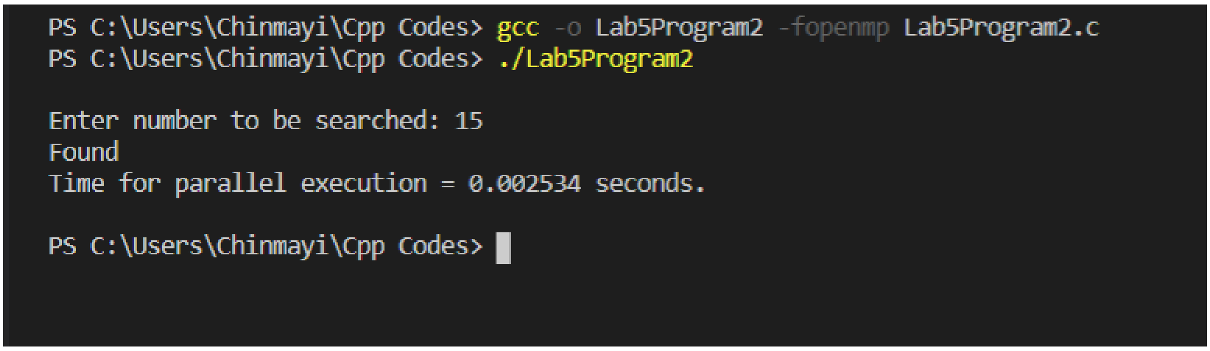
```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2 -fopenmp Lab5Program2.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 8765
Found
Time for parallel execution = 0.001362 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2seq -fopenmp Lab5Program2seq.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2seq

Enter number to be searched: 87654
Not found
Time for sequential execution = 0.000809 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

Array size = 10K.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2 -fopenmp Lab5Program2.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 8765
Found
Time for parallel execution = 0.002997 seconds.

PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2seq -fopenmp Lab5Program2seq.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2seq

Enter number to be searched: 56786
Not found
Time for sequential execution = 0.000719 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

Array size = 1Lakh.



```
PS C:\Users\Chinmayi\Cpp Codes> gcc -o Lab5Program2 -fopenmp Lab5Program2.c
PS C:\Users\Chinmayi\Cpp Codes> ./Lab5Program2

Enter number to be searched: 15
Found
Time for parallel execution = 0.002534 seconds.

PS C:\Users\Chinmayi\Cpp Codes> 
```

Array size = 10000000.