main.c

```c
#include <stdio.h>

void binary_search(int [], int, int, int);
void bubble_sort(int [], int);

int main()
{
    int key, size, i;
    int list[25];

    printf("Enter size of a list: ");
    scanf("%d", &size);
    printf("Enter elements\n");
    for(i = 0; i < size; i++)
    {
        scanf("%d",&list[i]);
    }
    bubble_sort(list, size);
    printf("\n");
    printf("Enter key to search\n");
    scanf("%d", &key);
    binary_search(list, 0, size, key);

}

void bubble_sort(int list[], int size)
{
    int temp, i, j;
    for (i = 0; i < size; i++)
```

main.c

```
27  {
28      int temp, i, j;
29      for (i = 0; i < size; i++)
30      {
31          for (j = i; j < size; j++)
32          {
33              if (list[i] > list[j])
34              {
35                  temp = list[i];
36                  list[i] = list[j];
37                  list[j] = temp;
38              }
39          }
40      }
41  }
42
43  void binary_search(int list[], int lo, int hi, int key)
44  {
45      int mid;
46
47      if (lo > hi)
48      {
49          printf("Key not found ");
50          return;
51      }
52      mid = (lo + hi) / 2;
53      if (list[mid] == key)
54      {
55          printf("Key found ");
```

main.c

```
46
47  if (lo > hi)
48  {
49      printf("Key not found\n");
50      return;
51  }
52  mid = (lo + hi) / 2;
53  if (list[mid] == key)
54  {
55      printf("Key found\n");
56  }
57  else if (list[mid] > key)
58  {
59      binary_search(list, lo, mid - 1, key);
60  }
61  else if (list[mid] < key)
62  {
63      binary_search(list, mid + 1, hi, key);
64  }
65  }
66
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
Enter size of a list: 6
Enter elements
6
1
2
3
5
6

Enter key to search
3
Key found
>
```

Page No.
Date: / /

Binary Search

```
#include <stdio.h>
Void binary_Search(int [], int, int, int);
Void bubble_sort (int [], int);

int main()
{
    int key, size, i;
    int list [25];

printf(" Enter size of a list : ");
 scanf("%d", &size);
 Printf("Enter elements to");
  for( i=0 ; i<size; i++)
{
   scanf("%d", &list[i]);
}
bubbleSort(list, size);
  printf("\n");
 printf("Enter key to Search to");
 Scanf("%d", &key);
  binary_Search(list, 0, size, key);
}
 Void bubble_Sort(int list[], int size)
{
 int temp, i, j;
    for(i=0; i<size; i++);

  for(j=i, j<size; j++)
```

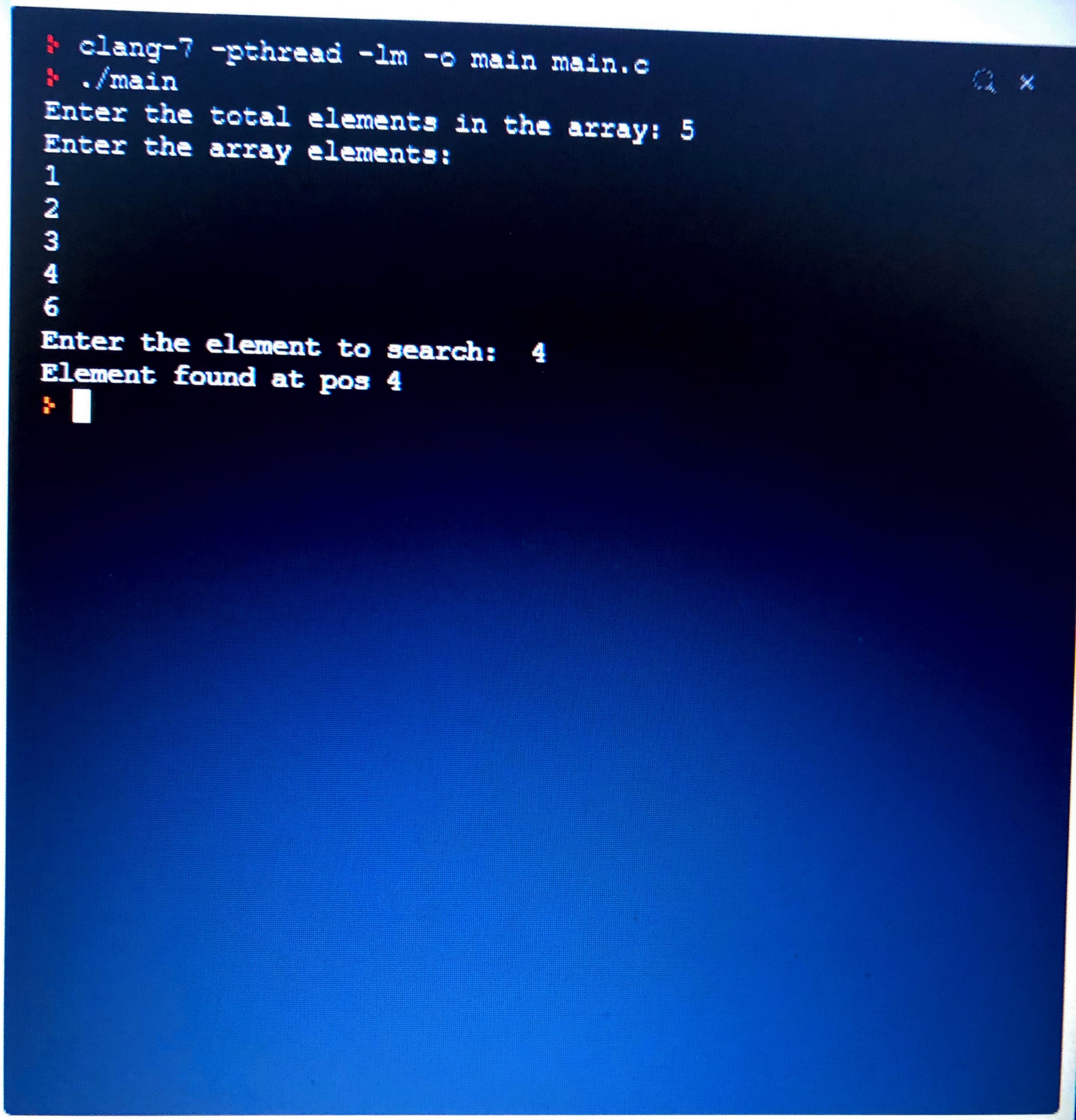
```
if (list[i] > list[j])
{
    temp = list[i];
    list[i] = list[j];
    list[j] = temp;
}
}
}
}

Void binarySearch(int list[], int lo, int hi,
                  int key)
{
int mid;
if (lo > hi)
{ printf("Key not found");
  return;
}
mid = (lo + hi) / 2;
if (list[mid] == key)
{
    printf("Key found \n");
if (lo > hi)
{ printf("Key not found \n");
  return;
}
mid = (lo + hi) / 2;
if (list[mid] == key)
{
    printf("Key found \n");
```

```
}
else if (list[mid] > key)
{
    binary_search(list, lo, mid - 1, key);
}
else if (list[mid] < key)
{
    binary_search(list, mid + 1, hi, key);
}
}
```

n.c

```c
#include <stdio.h>
int RecursiveLS(int arr[], int value, int index, int n)
{
    int pos = 0;

    if(index >= n)
    {
        return 0;
    }

    else if (arr[index] == value)
    {
        pos = index + 1;
        return pos;
    }

    else
    {
        return RecursiveLS(arr, value, index+1, n);
    }
    return pos;
}

int main()
{
    int n, value, pos, m = 0, arr[100];
    printf("Enter the total elements in the array: ");
    scanf("%d", &n);
```

```c
int main()
{
    int n, value, pos, m = 0, arr[100];
    printf("Enter the total elements in the array: ");
    scanf("%d", &n);

    printf("Enter the array elements: \n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }

    printf("Enter the element to search:  ");
    scanf("%d", &value);

    pos =  RecursiveLS(arr, value, 0, n);
    if (pos != 0)
    {
        printf("Element found at pos %d\n", pos);
    }
    else
    {
        printf("Element not found\n");
    }
    return 0;
}
```


clang-7 -pthread -lm -o main main.c
./main
Enter the total elements in the array: 5
Enter the array elements:
1
2
3
4
6
Enter the element to search: 4
Element found at pos 4

Page No.
Date : / /

Linear Search

```
#include<stdio.h>
int RecursiveLS(int arr[], int value, int index, int n)
{
    int pos = 0;
    if (index >= n)
    {
        return no;
    }
    else if (arr[index] == value)
    {
        pos = index + 1;
        return pos;
    }
    else
    {
        return RecursiveLS(arr, value, index+1, n);
    }
    return pos;
}
int main()
{
    int n, value, pos, no = 0, arr[100];
    printf("Enter the total element in the array:");
    scanf("%d", &n);

    printf("Enter the array elements:\n");
    for(int i = 0; i<n; i++)
    {
        scanf("%d", &arr[i]); }
```

Page No.
Date: / /

```
printf("Enter the element to Search: ");
scanf("%d", &value);

pos = RecursiveBS(arr, value, 0, n);
    if (pos != 0)
{
        printf("Element found at pos: %d\n", pos);
}
else
{
printf("Element not found\n");
}
return 0;
}
```