Lab Assingement – 8

Name : Rushang Bagada

Roll No : U24CS075

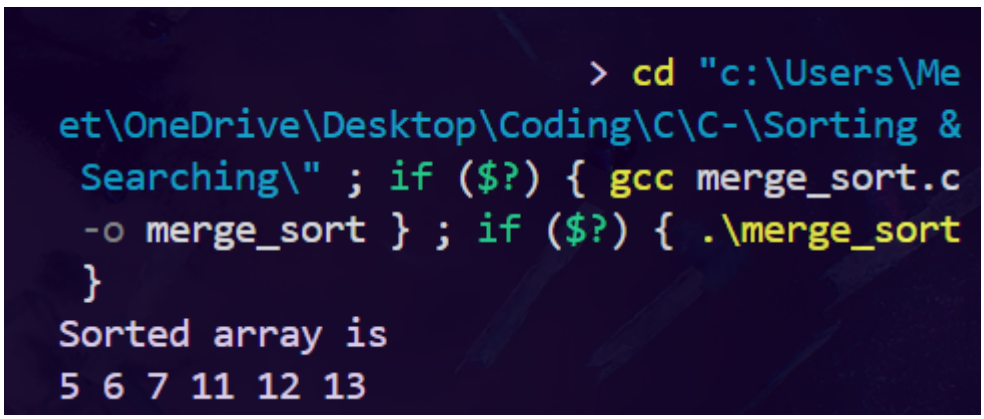
Q – 1

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


void merge(int arr[], int l, int m, int r) {
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (int i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    int i = 0, j = 0, k = l;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
```

```
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void merge_sort(int arr[],int l ,int r){
    if (l < r) {
        int m = l + (r - l) / 2;
        merge_sort(arr, l, m);
        merge_sort(arr, m + 1, r);
```

```
        merge(arr, l, m, r);
    }
}

int main (){
    int arr[] = {12, 11, 13, 5, 6, 7};
    int n = sizeof(arr)/sizeof(arr[0]);
    merge_sort(arr, 0, n - 1);
    printf("Sorted array is \n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
    return 0;
}
```

O/P:



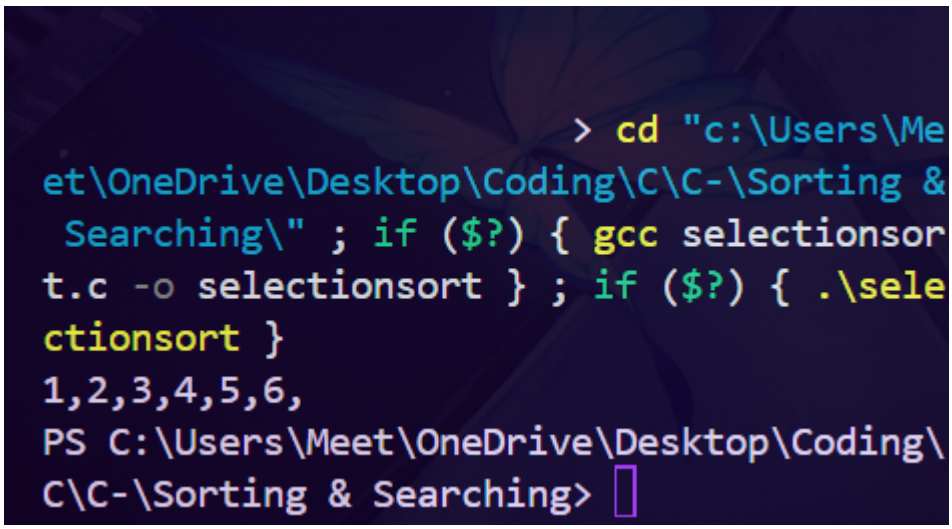
Q – 2

```
#include <stdio.h>
#include <limits.h>
 int n = 6;
```

```
void swap(int* a,int* b){
   int temp = *a ;
   *a = *b;
   *b = temp;
}

void selectionsort(int arr[]){
   for (int i=0;i<n;i++){
      int  a=i;
       int min = INT_MAX;
      for (int j=i;j<n;j++){
         if(arr[j]<min){
            min = arr[j];
            a = j;
         }
      }
      swap(&arr[i],&arr[a]);
   }
   return ;
}

void print(int arr[]){
   for (int i=0;i<n;i++){
      printf ("%d,",arr[i]);
   }
}
```

```
int main (){

    int arr[] = {4,5,1,3,2,6};
    selectionsort(arr);
    print(arr);
    return 0;
}
```

O/P:



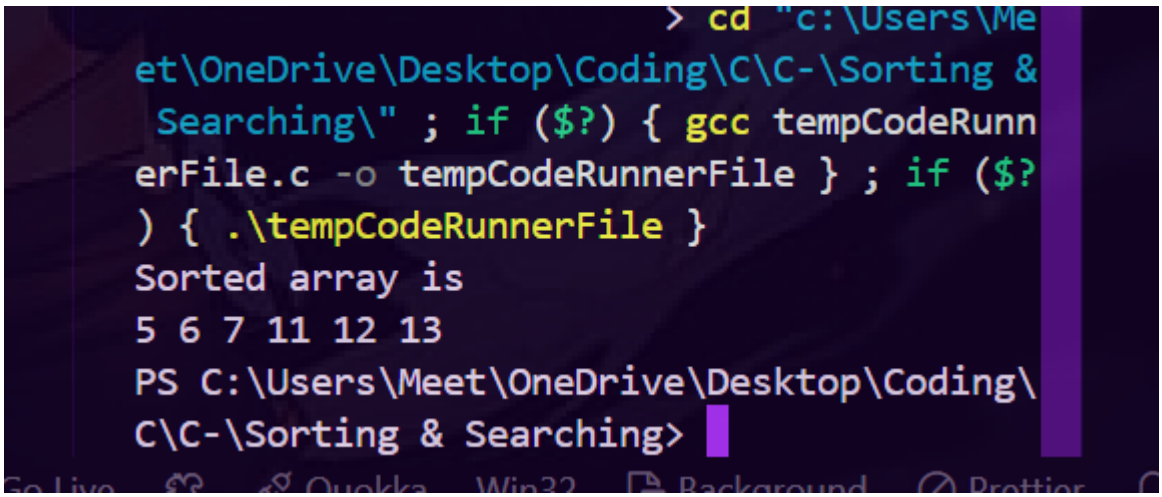
Q – 3

```
#include<stdio.h>

int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = (low - 1);
    for (int j = low; j < high; j++) {
        if (arr[j] < pivot) {
            i++;
            int temp = arr[i];
```

```
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    int temp = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp;
    return (i + 1);
}

void QuickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        QuickSort(arr, low, pi - 1);
        QuickSort(arr, pi + 1, high);
    }
}

int main(){
    int arr[] = {12, 11, 13, 5, 6, 7};
    int n = sizeof(arr)/sizeof(arr[0]);
    QuickSort(arr, 0, n - 1);
    printf("Sorted array is \n");
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
    printf("\n");
```

```
    return 0;
}
```

o/p:

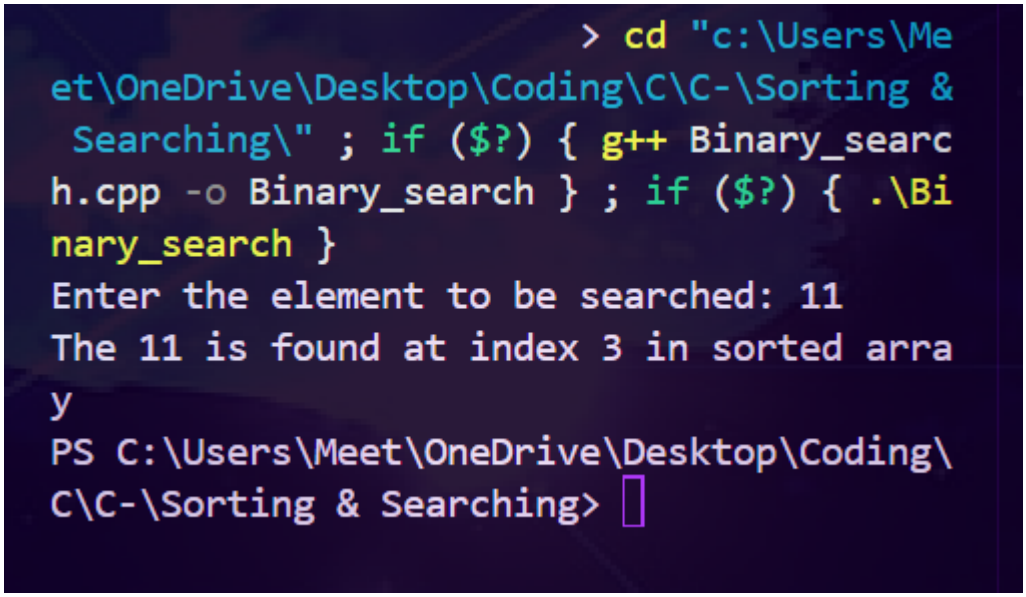


Q – 4

```
#include<iostream>
#include<algorithm>
#include<vector>

using namespace std;

int binary_search(vector<int>& arr, int a) {
    int left = 0, right = arr.size() - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (arr[mid] == a) {
            return mid;
        } else if (arr[mid] < a) {
```

```
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }
    return -1; // Element not found
}

int main(){
    vector<int> arr = {12, 11, 13, 5, 6, 7};
    sort(arr.begin(), arr.end());
    int a;
    cout<<"Enter the element to be searched: ";
    cin>>a;
    int index = binary_search(arr, a);
    if (index == -1) {
        cout << "Element not found" << endl;
        return 0;
    }
    cout<<"The " <<a<<" is found at index "<<index <<" in sorted array"<<endl;
    return 0;
}
```

O/P:


> cd "c:\Users\Meet\OneDrive\Desktop\Coding\C\C-\Sorting & Searching\" ; if ($?) { g++ Binary_search.cpp -o Binary_search } ; if ($?) { .\Binary_search }
Enter the element to be searched: 11
The 11 is found at index 3 in sorted array
PS C:\Users\Meet\OneDrive\Desktop\Coding\C\C-\Sorting & Searching>