Design And Analysis Of Algorithms

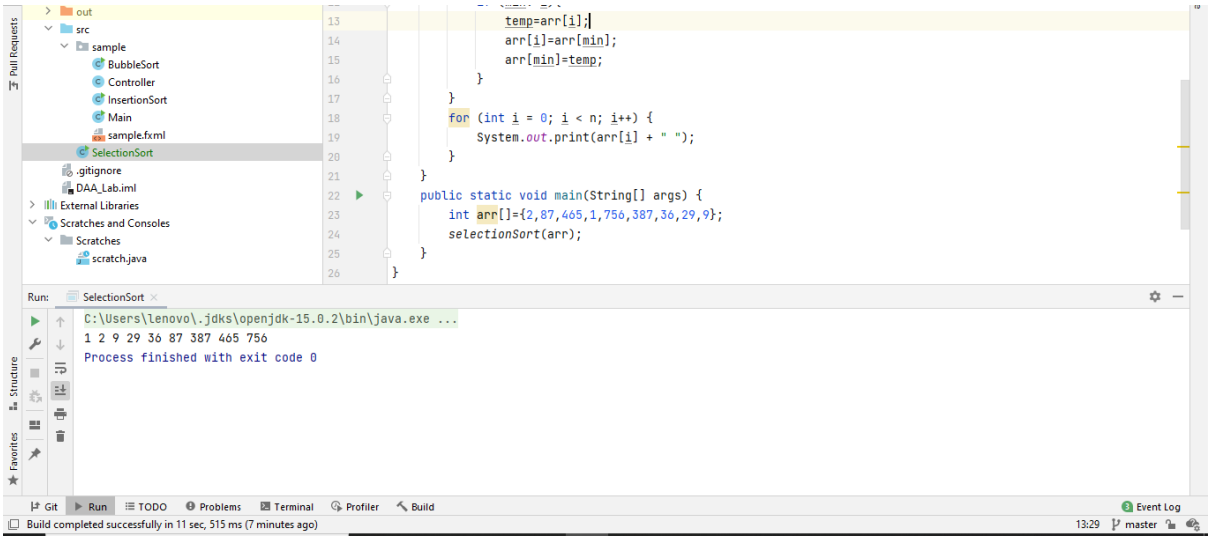
Practical- 2

Objective:-Implementation and analysis of Selection Sort.

Code:-

```
public class SelectionSort {
    public static void selectionSort(int arr[]){
        int temp;
        int n= arr.length;
        for (int i = 0; i < n-1; i++) {
            int min=i;
            for (int j =i+1; j <n ; j++) {
                if (arr[j]<arr[min]){
                    min=j;
                }
            }
            if (min!=i){
                temp=arr[i];
                arr[i]=arr[min];
                arr[min]=temp;
            }
        }
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
    }
    public static void main(String[] args) {
        int arr[]={2,87,465,1,756,387,36,29,9};
        selectionSort(arr);
    }
}
```

Output:-



Analysis:-

The best-case, Average and Worst Case time complexity of insertion sort is **O(n^2)**.

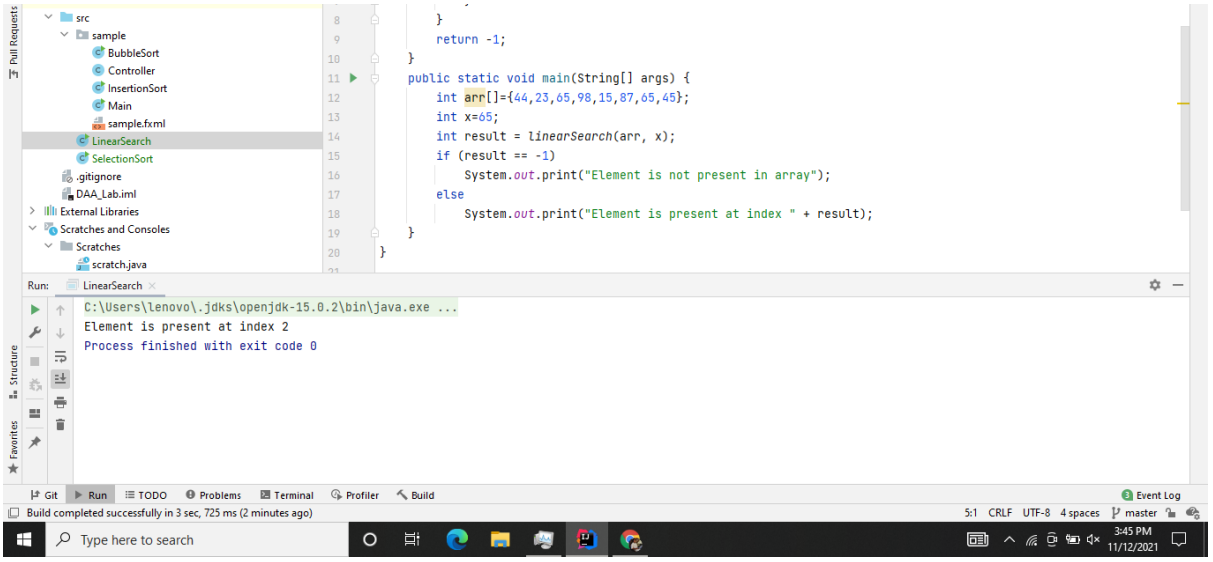
Practical- 2

Objective:-Implementation and analysis of **Linear Search**.

Code:-

```
public class LinearSearch {
    public static int linearSearch(int arr[], int x){
        int n= arr.length;
        for (int i = 0; i < n; i++) {
            if (arr[i]==x){
                return i;
            }
        }
        return -1;
    }
    public static void main(String[] args) {
        int arr[]={44,23,65,98,15,87,65,45};
        int x=65;
        int result = linearSearch(arr, x);
        if (result == -1)
            System.out.print("Element is not present in array");
        else
            System.out.print("Element is present at index " + result);
    }
}
```

Output:-



Analysis:-

The time complexity of Linear Search is O(n).

Linear search is rarely used practically because other search algorithms such as the binary search algorithm and hash tables allow significantly faster-searching comparison to Linear search.