

Name: Amaan Jamadar
MIS: 112103008
DSA-Assignment 4 & 5

Header Files:

1] Header file including the declaration of searching functions:

```
#ifndef SEARCHING_H
#define SEARCHING_H

int linear_search(int array[], int size, int target);
int binary_search(int array[] , int size, int target);

#endif
```

2] Header file including the declaration of sorting functions:

```
#ifndef SORTING_H
#define SORTING_H

int swap(int array[], int *a, int *b);
void bubble_sort(int array[] ,int size);
void selection_sort(int array[], int size);
void insertion_sort(int array[], int size);

#endif
```

Functions File:

```
#include<stdio.h>
int swap(int array[], int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
    return 0;
}

int linear_search(int array[], int size, int target){
    for(int i = 0; i < size ; i++){
        if(target == array[i]){
            printf("Target found at %d index.",i);
            return i;
        }
    }
    printf("Target not found.");
    return -1;
}
```

```

int binary_search(int array[] , int size, int target){
    int low , mid, high;
    low = 0;
    high = size -1;
    while(high >= low){
        mid = (high+low)/2;
        if(array[mid] == target){
            printf("Target found at %d", mid);
            return mid;
        }else if(array[mid] > target){
            high = mid -1;
        }else{
            low = mid + 1;
        }
    }
}

void bubble_sort(int array[] ,int size){
    int i , j ;
    for(i = 0 ; i < size; i++){
        for(j = 0; j< size-1-i; j++){
            if(array[j] > array[j+1]){
                swap(array, &array[j], &array[j+1]);
            }
        }
    }
}

```

```

void selection_sort(int array[], int size){
    int i , j, min ;
    for(i =0; i<size; i++){
        min = i;
        for(j = i ; j< size; j++){
            if(array[min]>array[j]){
                min = j;
            }
        }
        if(min != i){
            swap(array, &array[i], &array[min]);
        }
    }
}

void insertion_sort(int array[], int size){
    int i,j,temp ;
    for(i = 0; i<size; i++){
        temp = array[i];
        j = i-1;
        while(j>=0 && array[j]> temp){
            array[j+1] = array[j];
            j--;
        }
        array[j+1] = temp;
    }
}

```

Main File Code:

```
#include<stdio.h>
#include<stdlib.h>
#include"searching.h"
#include"sorting.h"

int main(){
    int n, target, ch, i ,j , k;

    printf("1- Linear Search\n");
    printf("2- Binary Search\n");
    printf("3- Selection Sort\n");
    printf("4- Bubble Sort\n");
    printf("5- Insertion Sort\n");
    printf("Enter Choice:");
    scanf("%d", &ch);
    if(ch == 1 || ch ==2){
        printf("Enter the number of elements of the array to be generated:");
        scanf("%d",&n);
        int array[n];
        for(i = 0; i <n;i++){
            printf("Enter the %d element",i);
            scanf("%d",&array[i]);
        }

        //printing the array
        for(j = 0; j <n ;j++){
            printf("%d ",array[j]);
        }

        //taking target
        printf("\nEnter the target to find :");
        scanf("%d",&target);
```

```
        if(ch == 1){
            linear_search(array, n , target);
        }
        else{
            binary_search(array, n , target);
        }
    }
    else if(ch == 3||ch == 4 || ch == 5 ){
        printf("Enter the number of elements of the array to be generated:");
        scanf("%d",&n);
        int array[n];
        // making array with random numbers from 1-100
        for(i = 0 ; i < n ;i++){
            array[i] = rand() %100 +1;
        }

        //printing the array
        for(j = 0; j <n ;j++){
            printf("%d ",array[j]);
        }

        if(ch == 3){
            //selection sort
            selection_sort(array,n);

            printf("\n Printing after Selection sort\n");
            //printing the array after selection sort
            for(j = 0; j <n ;j++){
                printf("%d ",array[j]);
            }
        }
    }
}
```

```

        else if(ch == 4){
            //bubble sorting
            bubble_sort(array , n);

            printf("\n Printing after Bubble sort\n");
            //printing the array after bubble sort
            for(j = 0; j <n ;j++){
                printf("%d ",array[j]);
            }
        }
        else{
            //insertion sorting
            insertion_sort(array , n);

            printf("\n Printing after insertion sort\n");
            //printing the array after insertion sort
            for(j = 0; j <n ;j++){
                printf("%d ",array[j]);
            }
        }
    }
}

```

Output:

1]

```

amaan@ubunx:~/Documents/DSA_Assign/DSA_ASSIGN4$ ls
a.out  binary_search.c  functions.c  main.c  searching.h  sorting.h
amaan@ubunx:~/Documents/DSA_Assign/DSA_ASSIGN4$ gcc main.c functions.c -o output.out
amaan@ubunx:~/Documents/DSA_Assign/DSA_ASSIGN4$ ./output.out
1- Linear Search
2- Binary Search
3- Selection Sort
4- Bubble Sort
5- Insertion Sort
Enter Choice:1
Enter the number of elements of the array to be generated:5
Enter the 0 element6
Enter the 1 element2
Enter the 2 element3
Enter the 3 element8
Enter the 4 element10
6 2 3 8 10
Enter the target to find :3
Target found at 2 index.

```

2]

```
amaan@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$ ./output.out
1- Linear Search
2- Binary Search
3- Selection Sort
4- Bubble Sort
5- Insertion Sort
Enter Choice:2
Enter the number of elements of the array to be generated:5
Enter the 0 element0
Enter the 1 element25
Enter the 2 element30
Enter the 3 element36
Enter the 4 element55
0 25 30 36 55
Enter the target to find :36
Target found at 3amaan@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$
```

3]

```
amaan@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$ ./output.out
1- Linear Search
2- Binary Search
3- Selection Sort
4- Bubble Sort
5- Insertion Sort
Enter Choice:3
Enter the number of elements of the array to be generated:10
84 87 78 16 94 36 87 93 50 22
Printing after Selection sort
16 22 36 50 78 84 87 87 93 94 aman@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$
```

4]

```
amaan@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$ ./output.out
1- Linear Search
2- Binary Search
3- Selection Sort
4- Bubble Sort
5- Insertion Sort
Enter Choice:4
Enter the number of elements of the array to be generated:10
84 87 78 16 94 36 87 93 50 22
Printing after Bubble sort
16 22 36 50 78 84 87 87 93 94 aman@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$
```

5]

```
amaan@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$ ./output.out
1- Linear Search
2- Binary Search
3- Selection Sort
4- Bubble Sort
5- Insertion Sort
Enter Choice:5
Enter the number of elements of the array to be generated:10
84 87 78 16 94 36 87 93 50 22
Printing after insertion sort
16 22 36 50 78 84 87 87 93 94 aman@ubunx:~/Documents/Dsa_Assign/Dsa_ASSIGN4$
```