TASK 1:

```
#include<iostream>
using namespace std;
bool insertAtindex(int arr[],int index, int
size, int data){
    int flag = 0;
    if(index > 10 && index < 0){
        cout<<"\n\tINDEX OUT OF RANGE!\n";
        return false;
    }
    for(int i=0; i<size; i++){
        if(i == index){
            arr[i] = data;
            cout<<data<<" INSERTED at index:
"<<i;
            flag = 1;
            break;
        }
    }
    if(flag == 0){
        return false;
    }
    else if(flag == 1){
        return true;
    }

}
```

```
void returnAtindex(int arr[],int index,int
size){
    int flag = 0;
    for(int i=0; i<size; i++){
        if(i == index){
            flag = 1;
            cout<<"\n\tThe value at index
"<<index<<" is: "<<arr[i]<<"\n";
        }
    }
    if(flag == 0){
        cout<<"\n\tThe value at index
"<<index<<" is: "<<-1<<"\n";
    }
}

bool search(int arr[],int size,int
searchData){
    int flag = 0;
    for(int i=0; i<size; i++){
        if(arr[i] == searchData){
            arr[i] = searchData;
            cout<<searchData<<" FOUNDED at
index: "<<i;
            flag = 1;
            break;
        }
```

```
    }
    if(flag == 0){
        cout<<searchData<<" NOT FOUND!";
        return false;
    }
    else if(flag == 1){
        return true;
    }

}

void sort(int arr[],int size){
    int temp = 0;
    for(int i = 0; i < size - 1; i++){
        for(int j = i+1; j<size; j++){
            if(arr[i]>arr[j]){
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = arr[i];
            }
        }
    }
}

void display (int arr[],int size){
    cout<<"\n\n\tArray after all the
operations is: \n";
    for(int i = 0; i< size; i++){
```

```
        cout<<arr[i]<<" ";
    }
}

int main(){
    int arr[10] = {0};
    cout<<"Enter elements of array:\n ";
    for(int i = 0; i<10; i++){
        cin>>arr[i];
    }
    cout<<"Array Elements are: ";
    for(int i = 0; i<10; i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    while(true){
        cout<<"\n\n";
        cout<<"0. EXIT PROGRAM\n";
        cout<<"1. Insert at Index\n";
        cout<<"2. Return at Index\n";
        cout<<"3. Search in array\n";
        cout<<"4. Sort array\n";
        cout<<"5. Display array\n";
        cout<<"Enter option: ";
        int option = 0, index = 0, data = 0,
searchData = 0;
        cin>>option;
        switch(option){
```

```
            case 0:
                cout<<"Program
Terminated\n\n";
                exit(0);
                break;
            case 1:
                index = 0, data = 0,
searchData = 0;
                cout<<"Enter data to INSERT:
";
                cin>>data;
                cout<<"Enter INDEX: ";
                cin>>index;
                insertAtindex(arr, index,
10, data);
                break;
            case 2:
                index = 0, data = 0,
searchData = 0;
                cout<<"Enter INDEX: ";
                cin>>index;
                returnAtindex(arr, index,
10);
                break;
            case 3:
                index = 0, data = 0,
searchData = 0;
```

```
                cout<<"Enter data to SEARCH:
";
                cin>>searchData;
                search(arr, 10, searchData);
                break;
            case 4:
                index = 0, data = 0,
searchData = 0;
                sort(arr, 10);
                cout<<"\n\tARRAY SORTED!\n";
                break;
            case 5:
                index = 0, data = 0,
searchData = 0;
                display(arr, 10);
                break;
            Default:
                cout<<"INVALID\n\n";
        }
    }


    return 0;
}
```

OUTPUTS:

```
Enter elements of array:
 1
66
3
4
8
9
5
2
98
4
Array Elements are: 1 66 3 4 8 9 5 2 98 4


0. EXIT PROGRAM
1. Insert at Index
2. Return at Index
3. Search in array
4. Sort array
5. Display array
Enter option: 1
Enter data to INSERT: 2
Enter INDEX: 999
 0. EXIT PROGRAM
 1. Insert at Index
 2. Return at Index
 3. Search in array
 4. Sort array
 5. Display array
 Enter option: 2
 Enter INDEX: 5

         The value at index 5 is: 9


 0. EXIT PROGRAM
 1. Insert at Index
 2. Return at Index
 3. Search in array
 4. Sort array
 5. Display array
 Enter option: 3
 Enter data to SEARCH: 98
 98 FOUNDED at index: 8

 0. EXIT PROGRAM
 1. Insert at Index
 2. Return at Index
 3. Search in array
 4. Sort array
 5. Display array
 Enter option: 4

         ARRAY SORTED!
```

```
 0. EXIT PROGRAM
 1. Insert at Index
 2. Return at Index
 3. Search in array
 4. Sort array
 5. Display array
 Enter option: 0
 Program Terminated
```

TASK 2:

```cpp
#include<iostream>
using namespace std;
void sortArray(int arr[],int size){
    int temp = 0;
```

```
    for(int i = 0; i < size; i++){
        for(int j = i+1; j < size; j++){
            temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
}

bool binarySearch(int arr[],int size,int data){
    sortArray(arr, size);
    int start = 0, end = size - 1, mid;
    while(start<=end){
        mid =end + (start - end) / 2;
        if(arr[mid] == data){
            return mid;
        }
        if(arr[mid] < data){
            start = mid + 1;
        }
        else if(arr[mid] > data){
            end = mid - 1;
        }
    }
    return -1;
}

int main(){
    int arr[10] = {0};
    cout<<"Enter elements of array:\n ";
    for(int i = 0; i<10; i++){
        cin>>arr[i];
    }
    cout<<"Array Elements are: ";
    for(int i = 0; i<10; i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    cout<<"Sorted array is: ";
    sortArray(arr, 10);
    for(int i = 0; i<10; i++){
        cout<<arr[i]<<" ";
    }
    cout<<"\nEnter the element to find: ";
    int data;
    cin>>data;
    int result = binarySearch(arr,10,data);
    if(result == -1){
```

```
        cout<<"NO ELEMENT FOUND!";
    }
    else{
        cout<<"ELEMENT IS FOUND!";
    }
    return 0;
}
```

Outputs:

```
Enter elements of array:
 10
9
8
7
6
5
4
3
2
1
Array Elements are: 10 9 8 7 6 5 4 3 2 1
Sorted array is: 1 2 3 4 5 6 7 8 9 10
Enter the element to find: 8
ELEMENT IS FOUND!
PS C:\Users\saadg\Downloads\CricketProVision>
```

Task 3:

```
#include <iostream>
using namespace std;
bool setArray(int arr[], int size = 10){
    bool flag = true;
    for (int i = 0; i < size-1; i++) {
        for (int j = i+1; j < size; j++) {
            if (arr[i] > arr[j]) {
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
    for (int i = 0; i < size-1; i++) {
        for (int j = i+1; j < size; j++) {
            if (arr[i] == arr[j]) {
                cout<<"\n\tNOT A SET!\n";
                return false;
            }
        }
    }
    if (flag==true){
        cout<<"\n\tARRAY is A SET!\n";
    }
```

```
}
int main(){
    int arr[10] = {0};
    cout<<"Enter elements of array:\n ";
    for(int i = 0; i<10; i++){
        cin>>arr[i];
    }
    cout<<"Array Elements are: ";
    for(int i = 0; i<10; i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    setArray(arr);

    return 0;
}
```

Outputs:

```
Enter elements of array:
 4
2
6
6
5
4
3
21
1
2
Array Elements are: 4 2 6 6 5 4 3 21 1 2

        NOT A SET!
Enter elements of array:
 1
2
3
4
5
6
7
8
9
10
Array Elements are: 1 2 3 4 5 6 7 8 9 10

        ARRAY is A SET!
PS C:\Users\saadg\Downloads\CricketProVision>
```

TASK 4:

```
#include <iostream>
using namespace std;
void LongestSubsequence(int arr[], int size = 10){
    int max = 0, count = 1;
```

```
    for(int i = 0; i < size; i++){
        if(arr[i]<=arr[i+1]){
            count++;
        }
        else{
            if(count>max){
                max = count;
                count = 1;
            }
        }
    }

    if (count > max) {
        max = count;
    }
    cout << max;
}
int main(){
    int arr[10] = {0};
    cout<<"Enter elements of array:\n ";
    for(int i = 0; i<10; i++){
        cin>>arr[i];
    }
    cout<<"Array Elements are: ";
    for(int i = 0; i<10; i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    LongestSubsequence(arr);

    return 0;
}
```

Outputs:

```
Enter elements of array:
 1
2
3
4
5
6
7
2
3
4
Array Elements are: 1 2 3 4 5 6 7 2 3 4
7
PS C:\Users\saadg\Downloads\CricketProVision>
```

```
Enter elements of array:
 1
2
3
4
2
3
4
5
6
1
Array Elements are: 1 2 3 4 2 3 4 5 6 1
5
PS C:\Users\saadg\Downloads\CricketProVision>
```

TASK 5:

```cpp
#include <iostream>
using namespace std;

double average_likes(int post_likes[], int totalPosts){

    double average = 0.0;
    int sum = 0;

    for(int i = 0; i < totalPosts; i++){
        sum = sum + post_likes[i];
    }

    average = (sum / totalPosts);

    return average;

}

int most_liked_post(int post_likes[], int totalPosts){

    int max = -999;

    for(int i = 0; i < totalPosts; i++){
        if(post_likes[i] > max) {
            max = post_likes[i];
        }
    }

    return max;

}

int least_liked_post(int post_likes[], int totalPosts){
```

```
    int min = 99999;

    for(int i = 0; i < totalPosts; i++){
        if(post_likes[i] < min) {
            min = post_likes[i];
        }
    }

    return min;

}

int main(){

    int totalPosts = 0, max = 0, min = 0;
    double  average = 0.0;

    cout<<"Enter total posts you have likes on: ";
    cin>>totalPosts;

    int* post_likes = new int[totalPosts];

    for(int i = 0; i < totalPosts; i++){
        cout<<"\nEnter total likes of Post "<<i+1<<": ";
        cin>>post_likes[i];
    }

    average = average_likes(post_likes, totalPosts);
    max = most_liked_post(post_likes, totalPosts);
    min = least_liked_post(post_likes, totalPosts);

    cout<<"\n\n->\tAVERAGE POST LIKES: "<<average;
    cout<<"\n\n->\tMOST LIKED POST: "<<max;
    cout<<"\n\n->\tLEAST LIKED POST: "<<min;

    return 0;
}
```

Outputs:

```
Enter total posts you have likes on: 10

Enter total likes of Post 1: 59

Enter total likes of Post 2: 65

Enter total likes of Post 3: 12

Enter total likes of Post 4: 15

Enter total likes of Post 5: 100

Enter total likes of Post 6: 23

Enter total likes of Post 7: 52

Enter total likes of Post 8: 11

Enter total likes of Post 9: 9

Enter total likes of Post 10: 125


->      AVERAGE POST LIKES: 47

->      MOST LIKED POST: 125

->      LEAST LIKED POST: 9
PS C:\Users\saadg\Downloads\CricketProVision>
```

```
Enter total posts you have likes on: 3

Enter total likes of Post 1: 1

Enter total likes of Post 2: 2

Enter total likes of Post 3: 3


->      AVERAGE POST LIKES: 2

->      MOST LIKED POST: 3

->      LEAST LIKED POST: 1
PS C:\Users\saadg\Downloads\CricketProVision>
```