

Lab5:

```
#include <iostream>
using namespace std;
const int SIZE = 10;
int sequentialSearch(int list[], int size, int item);
int binarySearch(int list[], int size, int item);
void bubbleSort(int arr[], int n);
void insertionSort(int arr[], int n);
void selectionSort(int arr[], int n);
int main(){
    int list[SIZE];
    int num;
    for(int i = 0; i < SIZE; i++){
        cout << "enter a number" << endl;
        cin >> num;
        list[i] = num;
    }
    int searchNum;
    cout << "enter the num you want to search(for binary search)" << endl;
    cin >> searchNum;
    int location = binarySearch(list, SIZE, searchNum);
    if(location == -1) cout << "there is not such an item in the list" <<
endl;
    else cout << "the location is " << location << endl;

    cout << "enter the num you want to search(for sequential search)" <<
endl;
    cin >> searchNum;
    location = sequentialSearch(list, SIZE, searchNum);
    if(location == -1) cout << "there is not such an item in the list" <<
endl;
    else cout << "the location is " << location << endl;

    int arr[] = {3, 5, 2, 8, 9};
    int arr2[] = {2, 5, 3, 3, 9};
    int arr3[] = {3, 7, 11, 9, 1};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    cout << "Sorted array: ";
    for(int i=0; i<n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
    selectionSort(arr2, n);
```

```

        cout << "Sorted array: ";
        for(int i=0; i<n; i++) {
            cout << arr2[i] << " ";
        }
        cout << endl;
        insertionSort(arr3, n);
        cout << "Sorted array: ";
        for(int i=0; i<n; i++) {
            cout << arr3[i] << " ";
        }
        cout << endl;
        return 0;
    }
}

int sequentialSearch(int list[], int size, int item){
    int loc = 0;
    bool found = false;
    while((loc < size) && (!found)){
        if(list[loc] == item) found = true;
        else loc++;
        if(found == true) return loc;
    }
    return -1;
}

int binarySearch(int list[], int size, int item){
    int left = 0;
    int right = size - 1;
    int mid = (left + right) / 2;
    while(left <= right){
        if(list[mid] == item) return mid;
        else if(list[mid] < item) left = mid + 1;
        else right = mid - 1;
        mid = (left + right) / 2;
    }
    return -1;
}

void bubbleSort(int arr[], int n) {
    for(int i=0; i<n-1; i++) {
        for(int j=0; j<n-i-1; j++) {
            if(arr[j] > arr[j+1]) {
                swap(arr[j], arr[j+1]);
            }
        }
    }
}

```

```

}
void selectionSort(int arr[], int n) {
    for(int i=0; i<n-1; i++) {
        int min_idx = i;
        for(int j=i+1; j<n; j++) {
            if(arr[j] < arr[min_idx]) {
                min_idx = j;
            }
        }
        swap(arr[i], arr[min_idx]);
    }
}
void insertionSort(int arr[], int n) {
    for(int i=1; i<n; i++) {
        int key = arr[i];
        int j = i-1;
        while(j >= 0 && arr[j] > key) {
            arr[j+1] = arr[j];
            j--;
        }
        arr[j+1] = key;
    }
}

```

Result:

```
enter a number
1
enter a number
3
enter a number
5
enter a number
7
enter a number
8
enter a number
9
enter a number
11
enter a number
13
enter a number
14
enter a number
17
enter the num you want to search(for binary search)
13
the location is 7
enter the num you want to search(for sequential search)
1
the location is 0
Sorted array: 2 3 5 8 9
Sorted array: 2 3 3 5 9
Sorted array: 1 3 7 9 11
```