- **BubbleSort**

```cpp
#include <bits/stdc++.h>
using namespace std;

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

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


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

void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        cout << arr[i] << " ";
    cout << endl;
}


int main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    cout<<"Sorted array: \n";
    printArray(arr, n);
    return 0;
}
```

OutPut

```
C:\Users\WALEED-ZAHID\Documents\bubbleSort.exe
Sorted array:
11 12 22 25 34 64 90

--------------------------------
Process exited after 0.01313 seconds with return value 0
Press any key to continue . . .
```

- **InsertionSort**

```cpp
#include <bits/stdc++.h>
using namespace std;


void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;


        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}


void printArray(int arr[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;
}


int main()
{
    int arr[] = { 12, 11, 13, 5, 6 };
    int n = sizeof(arr) / sizeof(arr[0]);

    insertionSort(arr, n);
    printArray(arr, n);

    return 0;
}
```

Output

```
C:\Users\WALEED-ZAHID\Documents\insertionSort.exe
5 6 11 12 13

--------------------------------
Process exited after 0.01286 seconds with return value 0
Press any key to continue . . .
```

- **SelectionSort**

```
 3
 4  void swap(int *xp, int *yp)
 5  {
 6      int temp = *xp;
 7      *xp = *yp;
 8      *yp = temp;
 9  }
10
11  void selectionSort(int arr[], int n)
12  {
13      int i, j, min_idx;
14
15      for (i = 0; i < n-1; i++)
16      {
17          min_idx = i;
18          for (j = i+1; j < n; j++)
19          if (arr[j] < arr[min_idx])
20              min_idx = j;
21
22          swap(&arr[min_idx], &arr[i]);
23      }
24  }
25
26  void printArray(int arr[], int size)
27  {
28      int i;
29      for (i=0; i < size; i++)
30          cout << arr[i] << " ";
31      cout << endl;
32  }
33
34  int main()
35  {
36      int arr[] = {64, 25, 12, 22, 11};
37      int n = sizeof(arr)/sizeof(arr[0]);
38      selectionSort(arr, n);
39      cout << "Sorted array: \n";
40      printArray(arr, n);
41      return 0;
42  }
```

OutPut

```
C:\Users\WALEED-ZAHID\Documents\selectionSort.exe
Sorted array:
11 12 22 25 64

--------------------------------
Process exited after 0.01106 seconds with return value 0
Press any key to continue . . .
```

- **Binary Search Tree**

```cpp
#include <bits/stdc++.h>
using namespace std;


int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;

        if (arr[mid] == x)
            return mid;


        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);


        return binarySearch(arr, mid + 1, r, x);
    }

    // We reach here when element is not
    // present in array
    return -1;
}

int main(void)
{
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = binarySearch(arr, 0, n - 1, x);
    (result == -1) ? cout << "Element is not present in array"
                   : cout << "Element is present at index " << result;
    return 0;
}

```

OutPut

```
C:\Users\WALEED-ZAHID\Documents\binarySerchTree.exe
Element is present at index 3
--------------------------------
Process exited after 0.01898 seconds with return value 0
Press any key to continue . . .
```