```
/*--------Bubble Sort( O(n^2) for all cases )--------*/

#include<iostream>
using namespace std;

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

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, j + 1);
            }
        }
    }
}

int main()
{
    int n;
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    bubbleSort(arr, n);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```

```cpp
/*-----Bubble Sort without Loop------*/

#include<iostream>
using namespace std;

void inputArray(int arr[], int n, int i)
{
    if(i < n)
    {
        cin >> arr[i];
        inputArray(arr, n, i + 1);
    }
}

void printArray(int arr[], int n, int i)
{
    if(i < n)
    {
        cout << arr[i] << " ";
        printArray(arr, n, i + 1);
    }
}


void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

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

int main()
{
    int n;
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    cout << "Enter the elements: \n";
    inputArray(arr, n, 0);

    bubbleSort(arr, n, 0);

    cout << "Sorted array is: \n";
    printArray(arr, n, 0);

    return 0;
}
```

```cpp
/*--------Counting Sort( O(n + k), where n is the---------
-----no. of elements & k is the range of input values---*/

#include<iostream>
using namespace std;

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void countingSort(int arr[], int n)
{
    int range = arr[0];
    for(int i = 1; i < n; i++)
    {
        range = max(range, arr[i]);
    }

    int countArr[10] = {0};
    for(int i = 0; i < n; i++)
    {
        countArr[arr[i]]++;
    }

    for(int i = 1; i <= range; i++)
    {
        countArr[i] += countArr[i - 1];
    }

    int outputArr[n];
    for(int i = n - 1; i >= 0; i--)
    {
        outputArr[--countArr[arr[i]]] = arr[i];
    }

    for(int i = 0; i < n; i++)
    {
        arr[i] = outputArr[i];
    }
}



int main()
{
    int n;
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    countingSort(arr, n);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```

```
/*--------Heap Sort( O(nlogn) for all cases )--------*/

#include<iostream>
using namespace std;

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

void heapify(int arr[], int n, int i)
{
    int left_child = 2 * i + 1;
    int right_child = 2 * i + 2;
    int large = i;

    if(left_child < n && arr[left_child] > arr[large])
    {
        large = left_child;
    }
    if(right_child < n && arr[right_child] > arr[large])
    {
        large = right_child;
    }

    if(large != i)
    {
        swap(arr, large, i);
        heapify(arr, n, large);
    }

}

void buildHeap(int arr[], int n)
{
    for(int i = (n / 2) - 1; i >= 0; i--)
    {
        heapify(arr, n, i);
    }
}

void heapSort(int arr[], int n)
{
    buildHeap(arr, n);
    for(int i = n - 1; i >= 0 ; i--)
    {
        swap(arr, i, 0);
        heapify(arr, i, 0);
    }
}

int main()
{
    int n;
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    heapSort(arr, n);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```

```cpp
/*--------Insertion Sort( O(n^2) for worst & average csaes O(n) for the

#include<iostream>
using namespace std;

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

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

int main()
{
    int n;
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    insertionSort(arr, n);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```

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

/*--------Merge Sort( O(nlogn) for all cases )--------*/

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

void merge(int arr[], int left, int mid, int right)
{
    int i = left;
    int j = mid + 1;
    int k = 0;
    int temparr[right - left + 1];

    while(i <= mid && j <= right)
    {
        if(arr[i] <= arr[j])
        {
            temparr[k] = arr[i];
            k++;
            i++;
        }
        else
        {
            temparr[k] = arr[j];
            k++;
            j++;
        }
    }
    if(i > mid)
    {
        while(j <= right)
        {
            temparr[k] = arr[j];
            j++;
            k++;
        }
    }
    else
    {
        while(i <= mid)
        {
            temparr[k] = arr[i];
            i++;
            k++;
        }
    }

    for(i = left; i <= right; i++)
    {
        arr[i] = temparr[i - left];
    }
}

void mergeSort(int arr[], int left, int right)
{
    if(left < right)
    {
        int mid = (right + left) / 2;

        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);

        merge(arr, left, mid, right);
    }
}

int main()
{
    int n;
```

```cpp
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    mergeSort(arr, 0, n - 1);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```

```cpp
/*--------Quick Sort( O(n^2) for worst case(rare), O(nlogn) for average

#include<iostream>
using namespace std;

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

int partition_(int arr[], int left, int right)
{
    int i = left;
    int j = right;
    int pivot = arr[left];

    while(i < j)
    {
        while(arr[i] <= pivot)
        {
            i++;
        }
        while(arr[j] > pivot)
        {
            j--;
        }

        if(i < j)
        {
            swap(arr, i, j);
        }

    }

    swap(arr, j, left);

    return j;

}

void quickSort(int arr[], int left, int right)
{
    if(left < right)
    {
        int partition = partition_(arr, left, right);

        quickSort(arr, left, partition - 1);
        quickSort(arr, partition + 1, right);
    }
}

int main()
{
    int n;
    cout << "Enter the length";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    quickSort(arr, 0, n - 1);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```

```cpp
/*--------Selection Sort( O(n^2) for all cases )--------*/

#include<iostream>
using namespace std;

void inputArray(int arr[], int n)
{
    cout << "Enter the elements of your array: \n";
    for(int i = 0; i < n; i++)
    {
        cin >> arr[i];
    }
}

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

void swap(int arr[], int i, int j)
{
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

void selectionSort(int arr[], int n)
{
    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)
        {
            swap(arr, i, min);
        }
    }
}

int main()
{
    int n;
    cout << "Enter the length: \n";
    cin >> n;

    int arr[n];
    inputArray(arr, n);

    selectionSort(arr, n);

    cout << "Sorted array is: \n";
    printArray(arr, n);

    return 0;
}
```