```cpp
/*--------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;
```

```
47:     cout << "Enter the length: \n";
48:     cin >> n;
49:
50:     int arr[n];
51:     inputArray(arr, n);
52:
53:     bubbleSort(arr, n);
54:
55:     cout << "Sorted array is: \n";
56:     printArray(arr, n);
57:
58:     return 0;
59: }
```

```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)
    {
```

```
47:         bubbleCompare(arr, n, i, 0);
48:         bubbleSort(arr, n, i + 1);
49:     }
50: }
51:
52: int main()
53: {
54:     int n;
55:     cout << "Enter the length: \n";
56:     cin >> n;
57:
58:     int arr[n];
59:     cout << "Enter the elements: \n";
60:     inputArray(arr, n, 0);
61:
62:     bubbleSort(arr, n, 0);
63:
64:     cout << "Sorted array is: \n";
65:     printArray(arr, n, 0);
66:
67:     return 0;
68: }
```

```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];
```

```
47:     }
48: 
49:     for(int i = 0; i < n; i++)
50:     {
51:         arr[i] = outputArr[i];
52:     }
53: }
54: 
55: 
56: 
57: int main()
58: {
59:     int n;
60:     cout << "Enter the length: \n";
61:     cin >> n;
62: 
63:     int arr[n];
64:     inputArray(arr, n);
65: 
66:     countingSort(arr, n);
67: 
68:     cout << "Sorted array is: \n";
69:     printArray(arr, n);
70: 
71:     return 0;
72: }
```

```cpp
/*--------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)
    {
```

```
47:         swap(arr, large, i);
48:         heapify(arr, n, large);
49:     }
50:
51: }
52:
53: void buildHeap(int arr[], int n)
54: {
55:     for(int i = (n / 2) - 1; i >= 0; i--)
56:     {
57:         heapify(arr, n, i);
58:     }
59: }
60:
61: void heapSort(int arr[], int n)
62: {
63:     buildHeap(arr, n);
64:     for(int i = n - 1; i >= 0 ; i--)
65:     {
66:         swap(arr, i, 0);
67:         heapify(arr, i, 0);
68:     }
69: }
70:
71: int main()
72: {
73:     int n;
74:     cout << "Enter the length: \n";
75:     cin >> n;
76:
77:     int arr[n];
78:     inputArray(arr, n);
79:
80:     heapSort(arr, n);
81:
82:     cout << "Sorted array is: \n";
83:     printArray(arr, n);
84:
85:     return 0;
86: }
```

```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()
{
```

```
47:     int n;
48:     cout << "Enter the length: \n";
49:     cin >> n;
50:
51:     int arr[n];
52:     inputArray(arr, n);
53:
54:     insertionSort(arr, n);
55:
56:     cout << "Sorted array is: \n";
57:     printArray(arr, n);
58:
59:     return 0;
60: }
```

```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
        {
```

```
47:             temparr[k] = arr[j];
48:             k++;
49:             j++;
50:         }
51:     }
52:     if(i > mid)
53:     {
54:         while(j <= right)
55:         {
56:             temparr[k] = arr[j];
57:             j++;
58:             k++;
59:         }
60:     }
61:     else
62:     {
63:         while(i <= mid)
64:         {
65:             temparr[k] = arr[i];
66:             i++;
67:             k++;
68:         }
69:     }
70:
71:     for(i = left; i <= right; i++)
72:     {
73:         arr[i] = temparr[i - left];
74:     }
75: }
76:
77: void mergeSort(int arr[], int left, int right)
78: {
79:     if(left < right)
80:     {
81:         int mid = (right + left) / 2;
82:
83:         mergeSort(arr, left, mid);
84:         mergeSort(arr, mid + 1, right);
85:
86:         merge(arr, left, mid, right);
87:     }
88: }
89:
90: int main()
91: {
92:     int n;
```

```
 93:     cout << "Enter the length: \n";
 94:     cin >> n;
 95:
 96:     int arr[n];
 97:     inputArray(arr, n);
 98:
 99:     mergeSort(arr, 0, n - 1);
100:
101:     cout << "Sorted array is: \n";
102:     printArray(arr, n);
103:
104:     return 0;
105: }
```

```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--;
        }

```

```
47:         if(i < j)
48:         {
49:             swap(arr, i, j);
50:         }
51:
52:     }
53:
54:     swap(arr, j, left);
55:
56:     return j;
57:
58: }
59:
60: void quickSort(int arr[], int left, int right)
61: {
62:     if(left < right)
63:     {
64:         int partition = partition_(arr, left, right);
65:
66:         quickSort(arr, left, partition - 1);
67:         quickSort(arr, partition + 1, right);
68:     }
69: }
70:
71: int main()
72: {
73:     int n;
74:     cout << "Enter the length";
75:     cin >> n;
76:
77:     int arr[n];
78:     inputArray(arr, n);
79:
80:     quickSort(arr, 0, n - 1);
81:
82:     cout << "Sorted array is: \n";
83:     printArray(arr, n);
84:
85:     return 0;
86: }
```

```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);
        }
    }
```

```
47: }
48:
49: int main()
50: {
51:     int n;
52:     cout << "Enter the length: \n";
53:     cin >> n;
54:
55:     int arr[n];
56:     inputArray(arr, n);
57:
58:     selectionSort(arr, n);
59:
60:     cout << "Sorted array is: \n";
61:     printArray(arr, n);
62:
63:     return 0;
64: }
```