

AQ1:

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

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

void selectionSort(int arr[], int n) {
    for(int i=0; i<n-1; i++) {
        int minIdx = i;
        for(int j=i+1; j<n; j++) {
            if(arr[j] < arr[minIdx])
                minIdx = j;
        }
        int temp = arr[minIdx];
        arr[minIdx] = arr[i];
        arr[i] = temp;
    }
}

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;
    }
}

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]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}
```

```

void merge(int arr[], int l, int m, int r) {
    int n1 = m-l+1;
    int n2 = r-m;
    int a[n1], b[n2];
    for(int i=0; i<n1; i++) a[i] = arr[l+i];
    for(int i=0; i<n2; i++) b[i] = arr[m+1+i];
    int i=0, j=0, k=l;
    while(i<n1 && j<n2) {
        if(a[i] <= b[j])
            arr[k++] = a[i++];
        else
            arr[k++] = b[j++];
    }
    while(i<n1) arr[k++] = a[i++];
    while(j<n2) arr[k++] = b[j++];
}

void mergeSort(int arr[], int l, int r) {
    if(l < r) {
        int m = (l+r)/2;
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);
        merge(arr, l, m, r);
    }
}

int partitionQ(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;
    for(int j=low; j<high; j++) {
        if(arr[j] < pivot) {
            i++;
            int t = arr[i];
            arr[i] = arr[j];
            arr[j] = t;
        }
    }
    int t = arr[i+1];
    arr[i+1] = arr[high];
    arr[high] = t;
    return i+1;
}

void quickSort(int arr[], int low, int high) {
    if(low < high) {
        int p = partitionQ(arr, low, high);
        quickSort(arr, low, p-1);
        quickSort(arr, p+1, high);
    }
}

```

```
    }
}

void copyArray(int a[], int b[], int n) {
    for(int i=0; i<n; i++) b[i] = a[i];
}

int main() {
    int n;
    cout << "Enter number of elements: ";
    cin >> n;

    int original[n];
    cout << "Enter elements: ";
    for(int i=0; i<n; i++) cin >> original[i];

    int arr[n];

    copyArray(original, arr, n);
    selectionSort(arr, n);
    cout << "Selection Sort: ";
    printArray(arr, n);

    copyArray(original, arr, n);
    insertionSort(arr, n);
    cout << "Insertion Sort: ";
    printArray(arr, n);

    copyArray(original, arr, n);
    bubbleSort(arr, n);
    cout << "Bubble Sort: ";
    printArray(arr, n);

    copyArray(original, arr, n);
    mergeSort(arr, 0, n-1);
    cout << "Merge Sort: ";
    printArray(arr, n);

    copyArray(original, arr, n);
    quickSort(arr, 0, n-1);
    cout << "Quick Sort: ";
    printArray(arr, n);

    return 0;
}
```

Output:

```
PS F:\Work\SEM3\DSA\LAB\7> cd "f:\Work\SEM3\DSA\LAB\7\" ; if ($?) { g++ AQ1.cpp -o AQ1 } ; if ($?) { ./AQ1 }
Enter number of elements: 6
Enter elements: 5
4
1
2
9
7
Selection Sort: 1 2 4 5 7 9
Insertion Sort: 1 2 4 5 7 9
Bubble Sort: 1 2 4 5 7 9
Merge Sort: 1 2 4 5 7 9
Quick Sort: 1 2 4 5 7 9
PS F:\Work\SEM3\DSA\LAB\7>
```

AQ2:

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

void improvedSelectionSort(int arr[], int n) {
    int left = 0, right = n - 1;

    while(left < right) {
        int minIdx = left, maxIdx = right;

        for(int i = left; i <= right; i++) {
            if(arr[i] < arr[minIdx]) minIdx = i;
            if(arr[i] > arr[maxIdx]) maxIdx = i;
        }

        int t = arr[left];
        arr[left] = arr[minIdx];
        arr[minIdx] = t;

        if(maxIdx == left) maxIdx = minIdx;

        t = arr[right];
        arr[right] = arr[maxIdx];
        arr[maxIdx] = t;

        left++;
        right--;
    }
}

int main() {
```

```
int n;
cout << "Enter number of elements: ";
cin >> n;
int arr[n];
cout << "Enter elements: ";
for(int i = 0; i < n; i++) cin >> arr[i];

improvedSelectionSort(arr, n);

for(int i = 0; i < n; i++) cout << arr[i] << " ";
return 0;
}
```

Output:

```
PS F:\Work\SEMB\DSA\LAB\7> cd "f:\Work\SEMB\DSA\LAB\7\" ; if ($?) { g++ AQ2.cpp -o AQ2 } ; if ($?) { .\AQ2 }
Enter number of elements: 5
Enter elements: 9
8
5
10
2
2 5 8 9 10
PS F:\Work\SEMB\DSA\LAB\7> []
```