ASSIGNMENT 7

NAME: Ujjwal Pant
ROLL NUMBER: 1024030370

Q1

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

// 1. Selection Sort
void selectionSort(vector<int> &a) {
    int n = a.size();
    for(int i = 0; i < n-1; i++) {
        int minIndex = i;
        for(int j = i+1; j < n; j++) {
            if(a[j] < a[minIndex]) minIndex = j;
        }
        swap(a[i], a[minIndex]);
    }
}

// 2. Insertion Sort
void insertionSort(vector<int> &a) {
    int n = a.size();
    for(int i = 1; i < n; i++) {
        int key = a[i];
        int j = i - 1;
        while(j >= 0 && a[j] > key) {
            a[j+1] = a[j];
```

```
            j--;
        }
        a[j+1] = key;
    }
}

// 3. Bubble Sort
void bubbleSort(vector<int> &a) {
    int n = a.size();
    for(int i = 0; i < n-1; i++) {
        for(int j = 0; j < n-i-1; j++) {
            if(a[j] > a[j+1]) swap(a[j], a[j+1]);
        }
    }
}

// 4. Merge Sort
void merge(vector<int> &a, int l, int m, int r) {
    vector<int> left(a.begin()+l, a.begin()+m+1);
    vector<int> right(a.begin()+m+1, a.begin()+r+1);

    int i=0, j=0, k=l;
    while(i < left.size() && j < right.size()) {
        if(left[i] <= right[j]) a[k++] = left[i++];
        else a[k++] = right[j++];
    }

    while(i < left.size()) a[k++] = left[i++];
    while(j < right.size()) a[k++] = right[j++];
}

void mergeSort(vector<int> &a, int l, int r) {
```

```
    if(l >= r) return;
    int m = (l+r)/2;
    mergeSort(a, l, m);
    mergeSort(a, m+1, r);
    merge(a, l, m, r);
}

// 5. Quick Sort
int partitionIdx(vector<int> &a, int l, int r) {
    int pivot = a[r];
    int i = l - 1;
    for(int j = l; j < r; j++) {
        if(a[j] < pivot) {
            i++;
            swap(a[i], a[j]);
        }
    }
    swap(a[i+1], a[r]);
    return i+1;
}

void quickSort(vector<int> &a, int l, int r) {
    if(l < r) {
        int pi = partitionIdx(a, l, r);
        quickSort(a, l, pi - 1);
        quickSort(a, pi + 1, r);
    }
}

int main() {
    vector<int> a = {64, 34, 25, 12, 22, 11, 90};
```

```
    cout << "Original Array: ";
    for(int x : a) cout << x << " ";
    cout << "\n";

    vector<int> b;

    b = a;
    selectionSort(b);
    cout << "Selection Sort: "; for(int x : b) cout << x << " "; cout << "\n";

    b = a;
    insertionSort(b);
    cout << "Insertion Sort: "; for(int x : b) cout << x << " "; cout << "\n";

    b = a;
    bubbleSort(b);
    cout << "Bubble Sort: "; for(int x : b) cout << x << " "; cout << "\n";

    b = a;
    mergeSort(b, 0, b.size()-1);
    cout << "Merge Sort: "; for(int x : b) cout << x << " "; cout << "\n";

    b = a;
    quickSort(b, 0, b.size()-1);
    cout << "Quick Sort: "; for(int x : b) cout << x << " "; cout << "\n";

    return 0;
}
```

Q2

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

// Improved Selection Sort – Picks min and max in each pass
void improvedSelectionSort(vector<int> &a) {
    int left = 0;
    int right = a.size() - 1;

    while(left < right) {
        int minIndex = left;
        int maxIndex = right;

        // Ensure correct ordering of indices initially
        if(a[minIndex] > a[maxIndex]) swap(a[minIndex], a[maxIndex]);

        for(int i = left+1; i < right; i++) {
            if(a[i] < a[minIndex]) minIndex = i;
            if(a[i] > a[maxIndex]) maxIndex = i;
        }

        swap(a[left], a[minIndex]);

        // If max element was at left, adjust its index
        if(maxIndex == left) maxIndex = minIndex;

        swap(a[right], a[maxIndex]);

        left++;
        right--;
    }
```

```
}

int main() {
    vector<int> a = {20, 5, 7, 1, 48, 30, 2};

    cout << "Original Array: ";
    for(int x : a) cout << x << " ";
    cout << "\n";

    improvedSelectionSort(a);

    cout << "Improved Selection Sort Result: ";
    for(int x : a) cout << x << " ";
    cout << "\n";

    return 0;
}
```