Name:-Dikesh Ganboi
Roll NO:- A36

```
#include <iostream>
#include <vector>
#include <chrono>

using namespace std;
using namespace std::chrono;

// Bubble Sort Algorithm
void bubbleSort(vector<int>& arr) {
    int n = arr.size();
    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], arr[j + 1]);
            }
        }
    }
}

// Insertion Sort Algorithm
void insertionSort(vector<int>& arr) {
    int n = arr.size();
    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 = j - 1;
        }
        arr[j + 1] = key;
    }
}

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

// Function to print array
void printArray(const vector<int>& arr) {
    for (int num : arr) {
        cout << num << " ";
    }
    cout << endl;
}

// Function to measure execution time of sorting algorithms
void measureTime(const string& sortName, void (*sortFunc)(vector<int>&),
vector<int>& arr) {
    auto start = high_resolution_clock::now();
    sortFunc(arr);
    auto stop = high_resolution_clock::now();
    auto duration = duration_cast<microseconds>(stop - start);
    cout << sortName << " Time: " << duration.count() << " microseconds" <<
endl;
}

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

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

    cout << "Original array: ";
    printArray(arr);

    cout << endl;

    cout << "Sorting Algorithms Time Complexity:" << endl;
    measureTime("Bubble Sort", bubbleSort, arr);
    measureTime("Insertion Sort", insertionSort, arr);
    measureTime("Selection Sort", selectionSort, arr);

    cout << endl;
```

```cpp
    cout << "Sorted array using Bubble Sort: ";
    printArray(arr);

    return 0;
}
```

OUTPUT:-

```
PS C:\Users\HP\Desktop\DAA EXperiment> cd "c:\Users\HP\Desktop\DAA
EXperiment\" ; if ($?) { g++ second.cpp -o second } ; if ($?) {
.\second }

Enter the number of elements: 6
Enter 6 elements: 34 12 15 19 22 15
Original array: 34 12 15 19 22 15

Sorting Algorithms Time Complexity:
Bubble Sort Time: 0 microseconds
Insertion Sort Time: 0 microseconds
Selection Sort Time: 0 microseconds

Sorted array using Bubble Sort: 12 15 15 19 22 34
```