Assignment-2 (DAA)

- Name: KRISHNA PANDEY
- Roll No.: B110
- Admission No.: U20CS110

1.1. Implement the above algorithms using the programming language of your choice.

I chose C++ to implement the above algorithms.

1.2. Provide the details of Hardware/Software you used to implement algorithms and to measure the time.

Hardware details is provided below and the IDE used by me is Visual Studio.

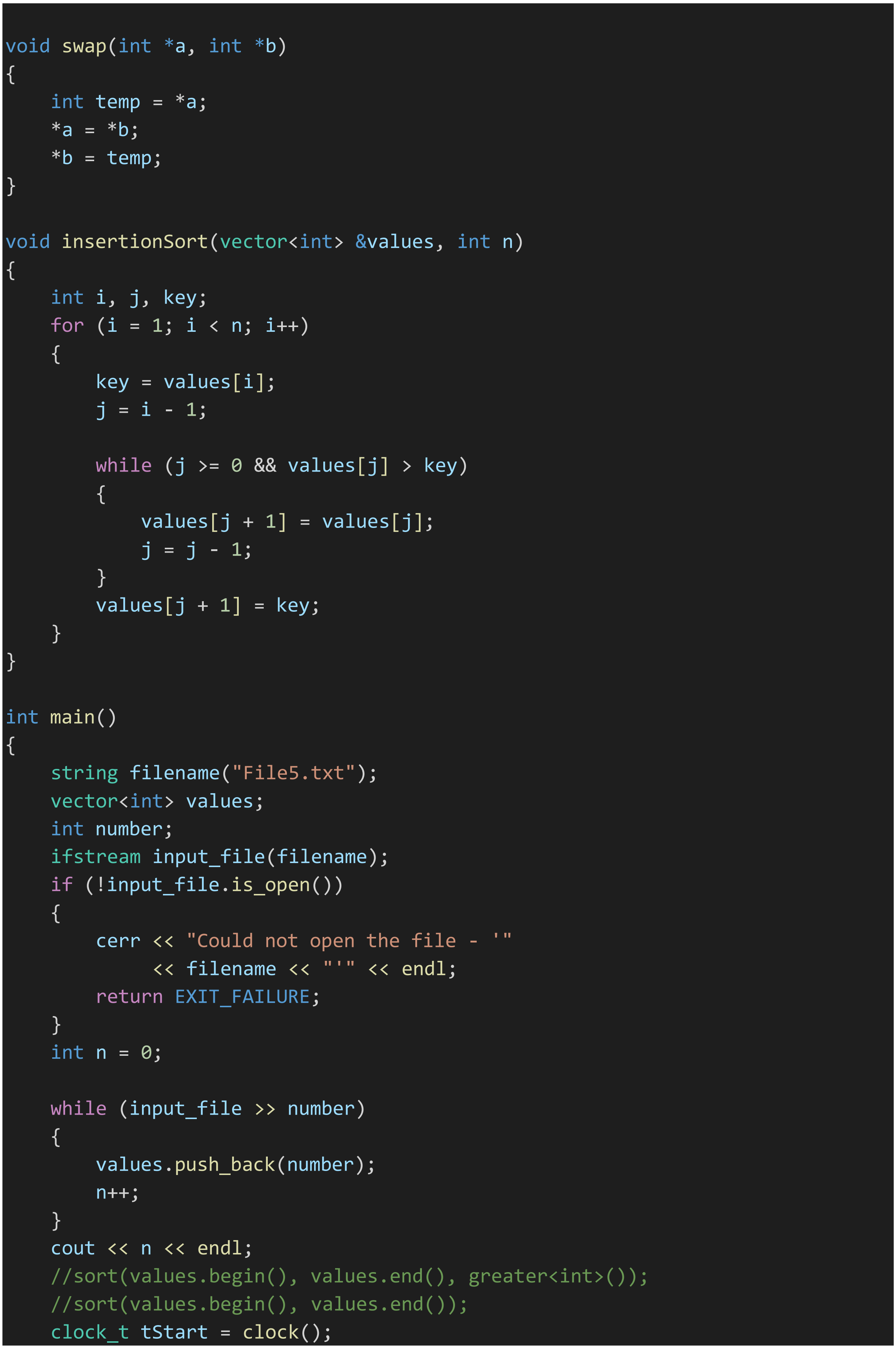
Device name	LAPTOP-LEHQUDIU
Processor	11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.42 GHz
Installed RAM	8.00 GB (7.75 GB usable)
Device ID	2C14C1C1-1420-41A4-B7F5-0D78ADA245F8
Product ID	00327-36277-89595-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

Edition	Windows 10 Home Single Language
Version	21H1
Installed on	29-07-2021
OS build	19043.1526
Experience	Windows Feature Experience Pack 120.2212.4170.0

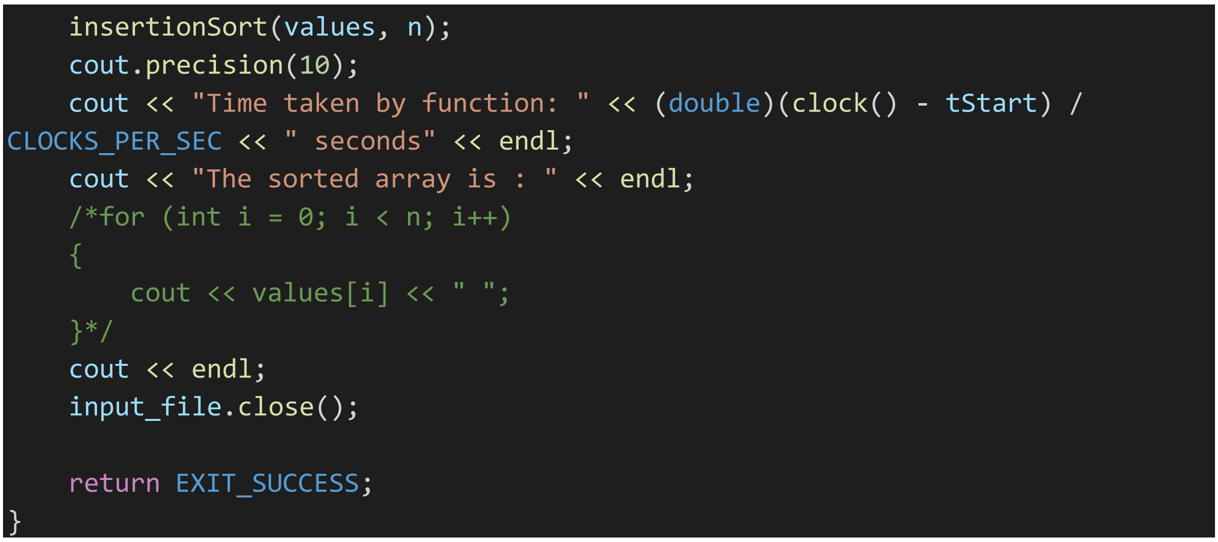
1.3. Submit the code (complete programs).

Insertion Sort:

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

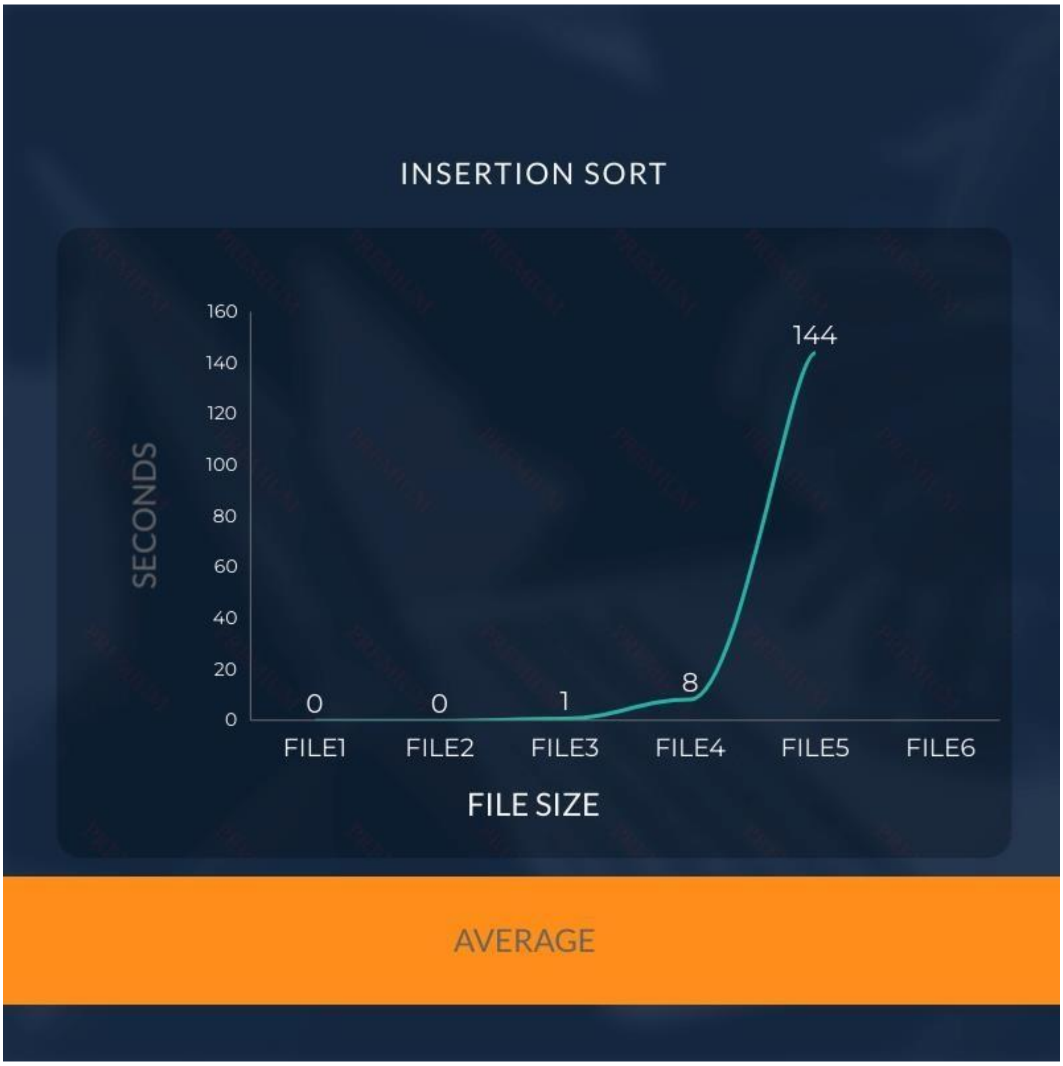


```
void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void insertionSort(vector<int> &values, int n)
{
    int i, j, key;
    for (i = 1; i < n; i++)
    {
        key = values[i];
        j = i - 1;

        while (j >= 0 && values[j] > key)
        {
            values[j + 1] = values[j];
            j = j - 1;
        }
        values[j + 1] = key;
    }
}

int main()
{
    string filename("File5.txt");
    vector<int> values;
    int number;
    ifstream input_file(filename);
    if (!input_file.is_open())
    {
        cerr << "Could not open the file - '"
             << filename << "'" << endl;
        return EXIT_FAILURE;
    }
    int n = 0;

    while (input_file >> number)
    {
        values.push_back(number);
        n++;
    }
    cout << n << endl;
    //sort(values.begin(), values.end(), greater<int>());
    //sort(values.begin(), values.end());
    clock_t tStart = clock();
```

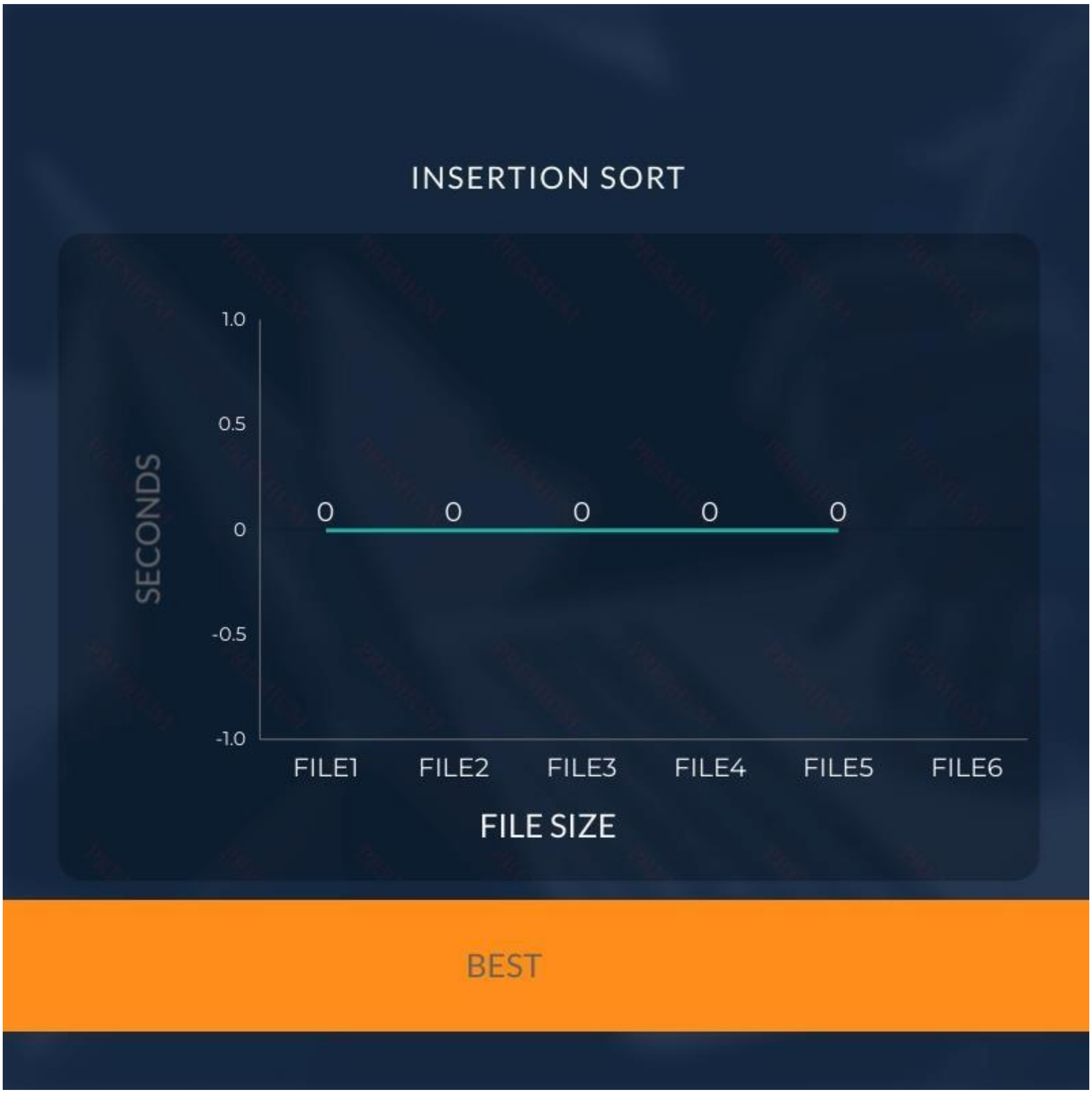


```
    insertionSort(values, n);
    cout.precision(10);
    cout << "Time taken by function: " << (double)(clock() - tStart) /
CLOCKS_PER_SEC << " seconds" << endl;
    cout << "The sorted array is : " << endl;
    /*for (int i = 0; i < n; i++)
    {
        cout << values[i] << " ";
    }*/
    cout << endl;
    input_file.close();

    return EXIT_SUCCESS;
}
```

1.4. Measure the average-case time (considering current data of six files) of insertion sort for all six files. Plot a graph.

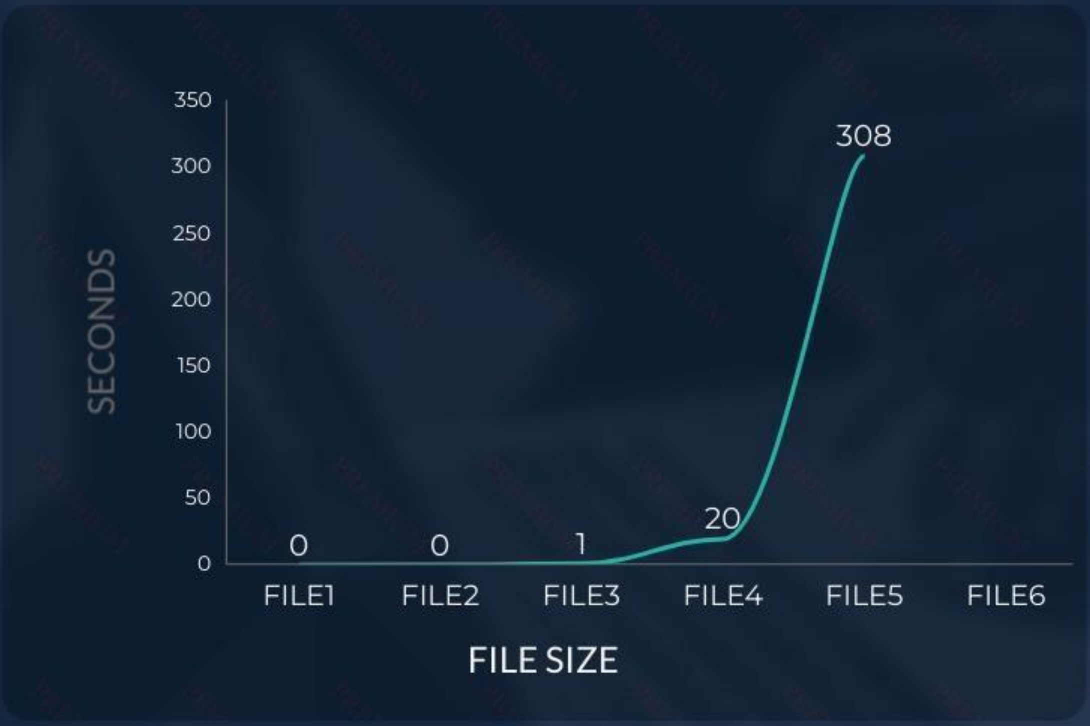


1.5 Measure the best-case time of insertion sort for all six files. Plot a graph.



1.6. Measure the worst-case time of insertion sort for all six files. Plot a graph.

INSERTION SORT



WORST