LAB WORK - 3

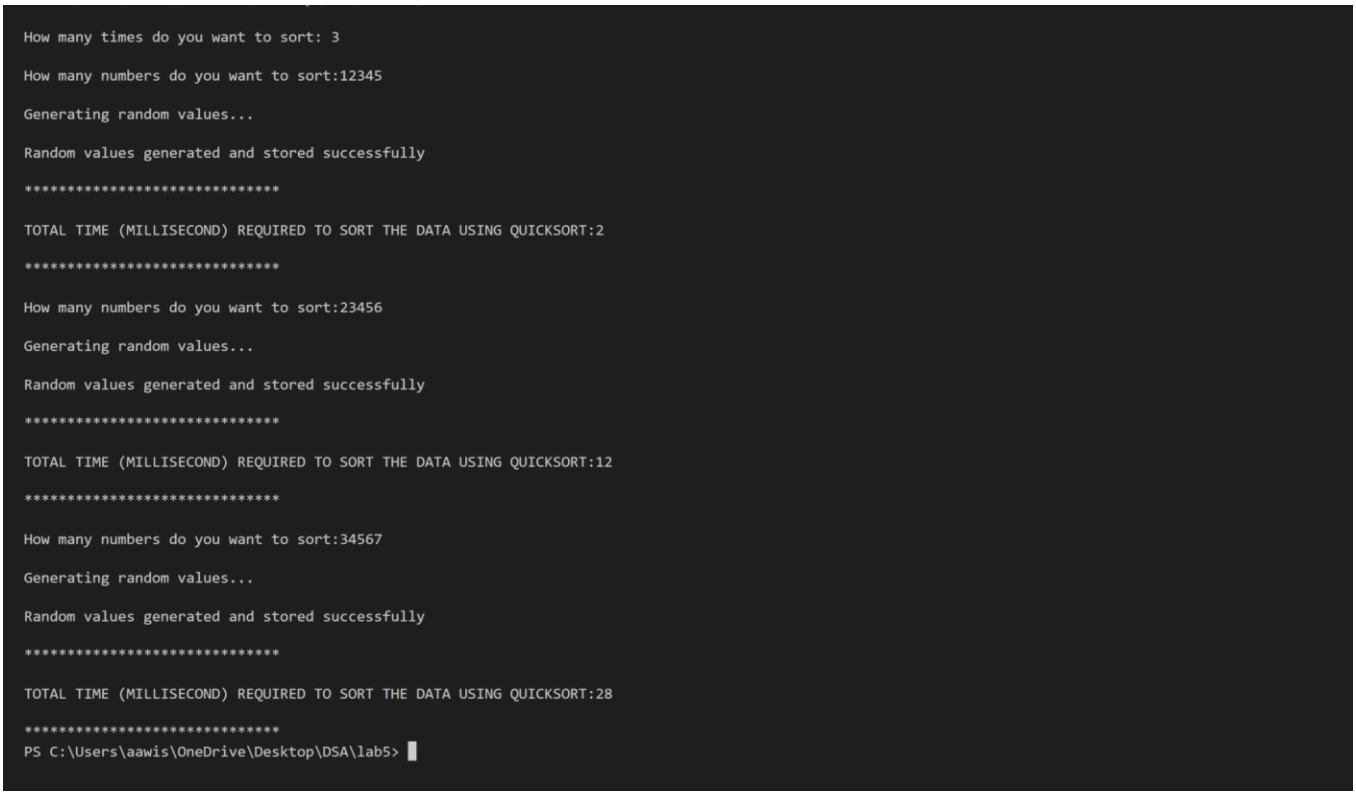
Aawishkar Tiwari

Computer Engineering – 3rd Semester

Roll no – 59

GitHub: https://github.com/SamirWagle/CE2020_Lab5_59_60

Output Screen:



Sort.cpp

For Quicksort:

```cpp
#include <vector>

int partition(std::vector<int> & vect, int small, int large)
{
    int pivot = vect[large];
    int i = small-1;
    for(int j=small;j<large;j++){
        if(vect[j]<=pivot){
            i=i+1;
            int temp1=vect[i];
            vect[i]=vect[j];
```

```
            vect[j]=temp1;
        }
    }

    int temp2 = vect[i+1];
    vect[i+1] = vect[large];
    vect[large] = temp2;
    return i+1;
}

void QuickSort(std::vector<int> & vect, int low, int high)
{
    int pivot;

    if (low < high)
    {
        pivot = partition(vect, low, high);
        QuickSort(vect, low, pivot - 1);
        QuickSort(vect, pivot + 1, high);
    }
}
```

Main.cpp

For QuickSort

```
#include <iostream>
#include <vector>
#include <ctime>
#include <random>
#include <chrono>
#include "sort.cpp"

using namespace std::chrono;
using namespace std;

int main()
{
    int size = 0, times;
    string choice;
    bool repeat = true;

    cout << "\nHow many times do you want to sort: ";
    cin >> times;
```

```
    srand(time(nullptr));

    for (int i = 1; i <= times; i++)
    {
        std::vector<int> vector1;
        std::vector<int> vector2;

        cout << "\nHow many numbers do you want to sort:";
        cin >> size;

        cout << "\nGenerating random values..." << endl;
        for (int j = 0; j < size; j++)
        {
            int value = rand() % 500;
            vector1.push_back(value);
            vector2.push_back(value);
        }
        cout << "\nRandom values generated and stored successfully" << endl;

        auto startquick = high_resolution_clock::now();
        QuickSort(vector1, 0, size);
        auto stopquick = high_resolution_clock::now();
        auto durationquick = duration_cast<milliseconds>(stopquick - startquick);

        cout << "\n*****************************" << endl;

        cout << "\nTOTAL TIME (MILLISECOND) REQUIRED TO SORT THE DATA USING
QUICKSORT:" << durationquick.count() << endl;

        cout << "\n*****************************" << endl;
    }
}
```