EUROPEAN UNIVERSITY OF LEFKE
Faculty of Engineering
Department of Computer Engineering

COMP218

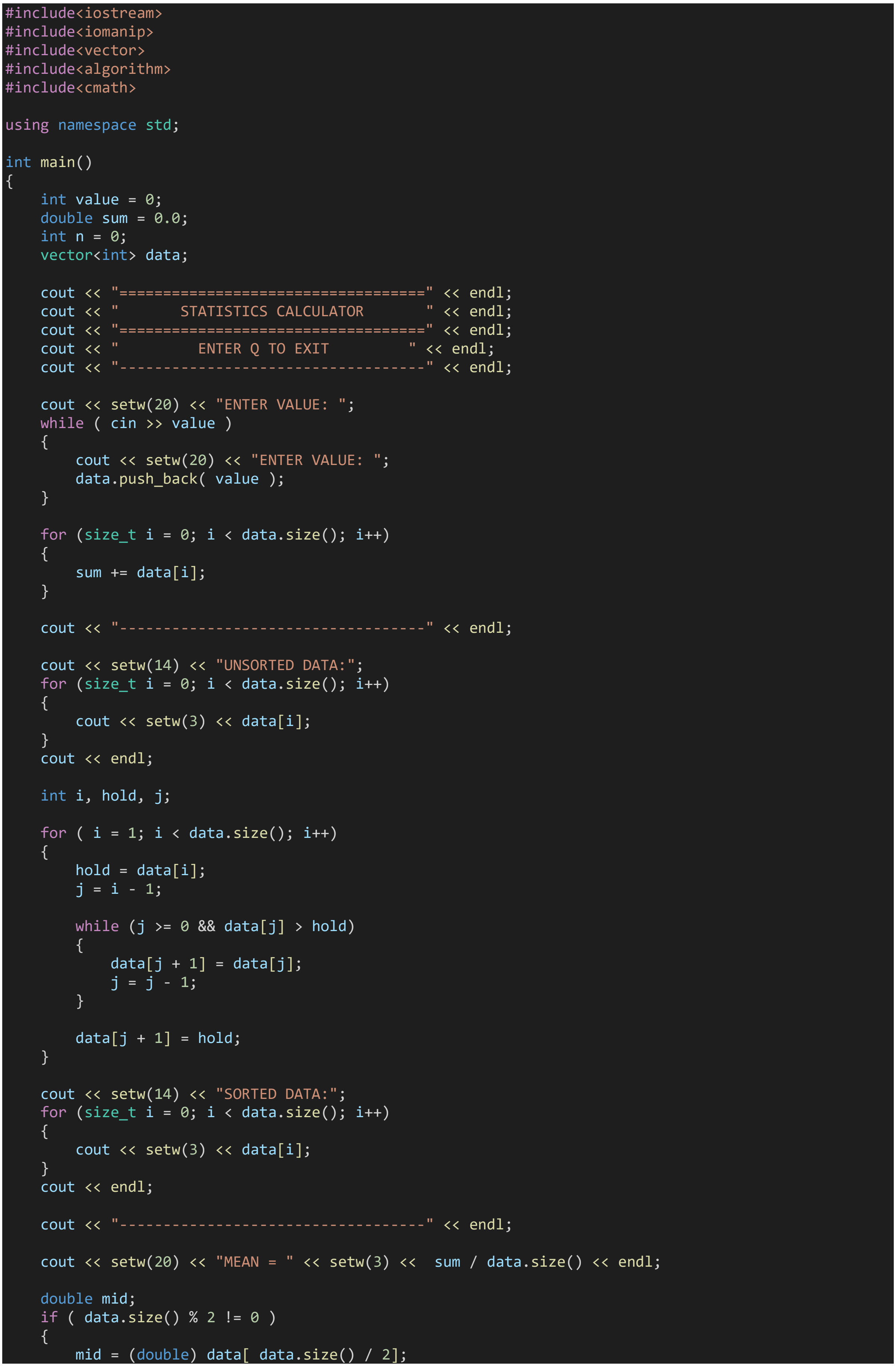
OBJECT-ORIENTED PROGRAMMING

Lab Work No. 4

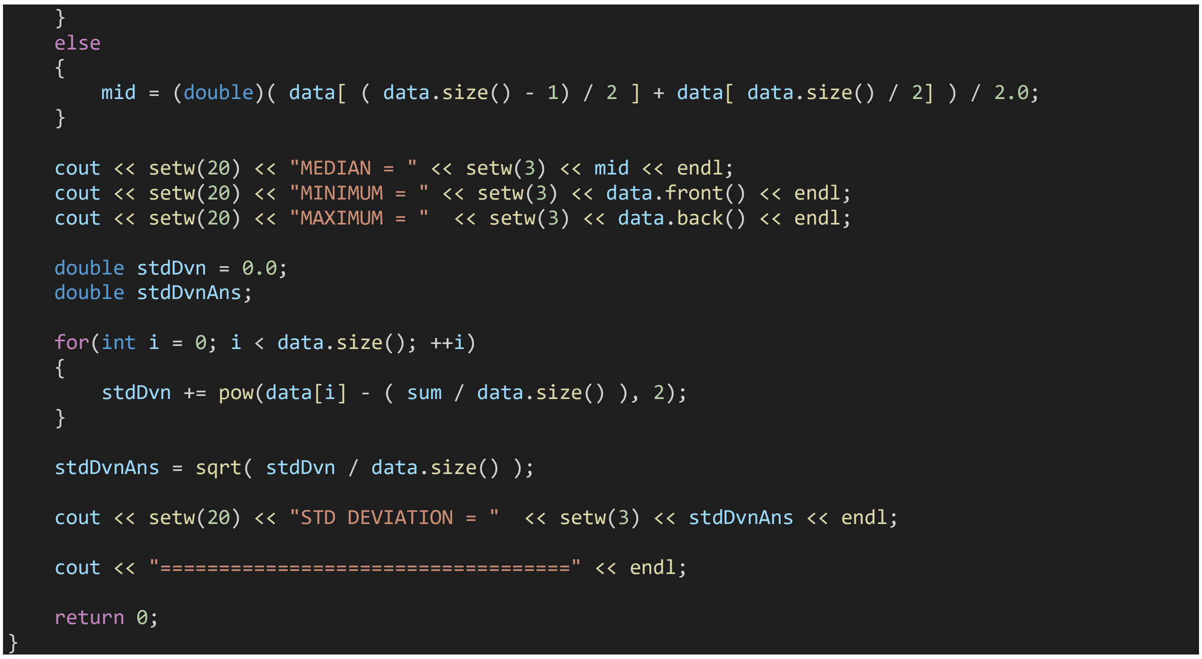
Prepared by Seward Richard Mupereri (20140175)

Submitted to Dr. Ferhun Yorgancıoğlu

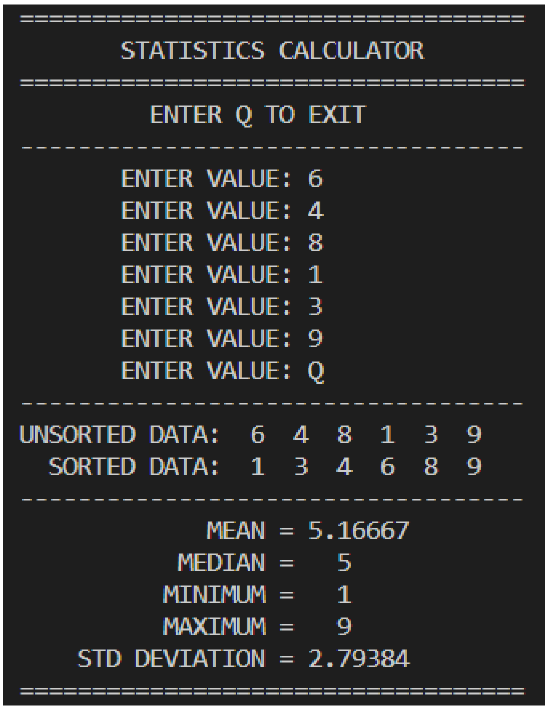
Task (1)



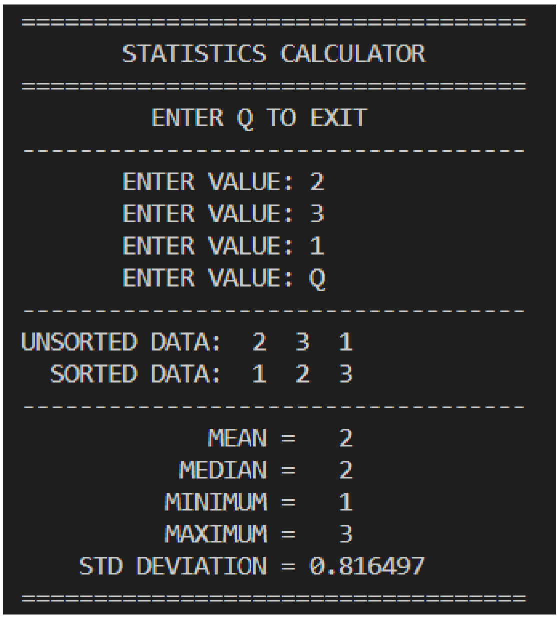
```
#include<iostream>
#include<iomanip>
#include<vector>
#include<algorithm>
#include<cmath>

using namespace std;

int main()
{
    int value = 0;
    double sum = 0.0;
    int n = 0;
    vector<int> data;

    cout << "===================================" << endl;
    cout << "       STATISTICS CALCULATOR       " << endl;
    cout << "===================================" << endl;
    cout << "         ENTER Q TO EXIT         " << endl;
    cout << "-----------------------------------" << endl;

    cout << setw(20) << "ENTER VALUE: ";
    while ( cin >> value )
    {
        cout << setw(20) << "ENTER VALUE: ";
        data.push_back( value );
    }

    for (size_t i = 0; i < data.size(); i++)
    {
        sum += data[i];
    }

    cout << "-----------------------------------" << endl;

    cout << setw(14) << "UNSORTED DATA:";
    for (size_t i = 0; i < data.size(); i++)
    {
        cout << setw(3) << data[i];
    }
    cout << endl;

    int i, hold, j;

    for ( i = 1; i < data.size(); i++)
    {
        hold = data[i];
        j = i - 1;

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

        data[j + 1] = hold;
    }

    cout << setw(14) << "SORTED DATA:";
    for (size_t i = 0; i < data.size(); i++)
    {
        cout << setw(3) << data[i];
    }
    cout << endl;

    cout << "-----------------------------------" << endl;

    cout << setw(20) << "MEAN = " << setw(3) <<  sum / data.size() << endl;

    double mid;
    if ( data.size() % 2 != 0 )
    {
        mid = (double) data[ data.size() / 2];
```



```cpp
    }
    else
    {
        mid = (double)( data[ ( data.size() - 1) / 2 ] + data[ data.size() / 2] ) / 2.0;
    }

    cout << setw(20) << "MEDIAN = " << setw(3) << mid << endl;
    cout << setw(20) << "MINIMUM = " << setw(3) << data.front() << endl;
    cout << setw(20) << "MAXIMUM = "  << setw(3) << data.back() << endl;

    double stdDvn = 0.0;
    double stdDvnAns;

    for(int i = 0; i < data.size(); ++i)
    {
        stdDvn += pow(data[i] - ( sum / data.size() ), 2);
    }

    stdDvnAns = sqrt( stdDvn / data.size() );

    cout << setw(20) << "STD DEVIATION = "  << setw(3) << stdDvnAns << endl;

    cout << "===================================" << endl;

    return 0;
}
```



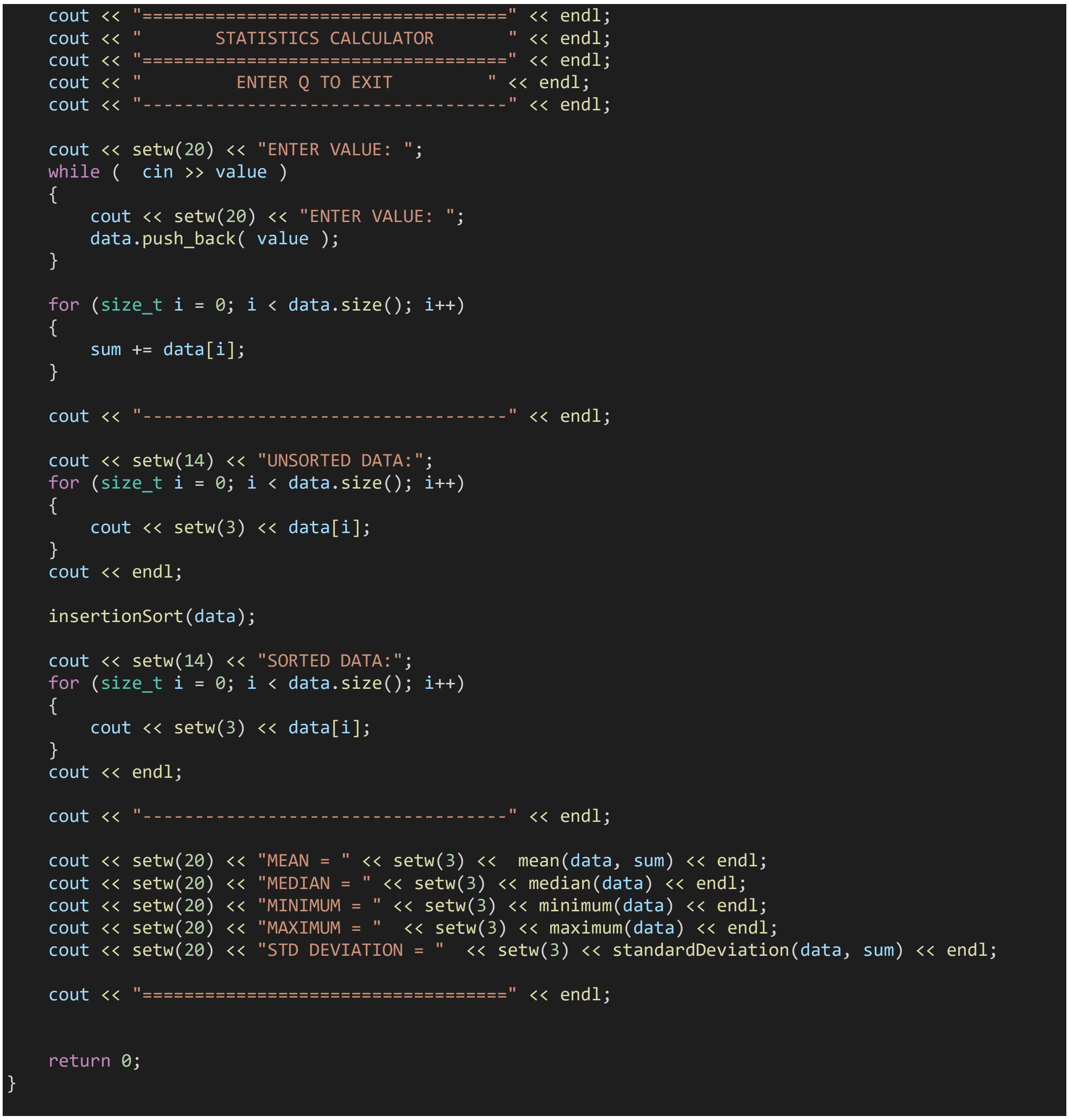
```
===================================
       STATISTICS CALCULATOR
===================================
         ENTER Q TO EXIT
-----------------------------------
       ENTER VALUE: 6
       ENTER VALUE: 4
       ENTER VALUE: 8
       ENTER VALUE: 1
       ENTER VALUE: 3
       ENTER VALUE: 9
       ENTER VALUE: Q
-----------------------------------
UNSORTED DATA:   6  4  8  1  3  9
  SORTED DATA:   1  3  4  6  8  9
-----------------------------------
             MEAN = 5.16667
           MEDIAN =   5
          MINIMUM =   1
          MAXIMUM =   9
    STD DEVIATION = 2.79384
===================================
```



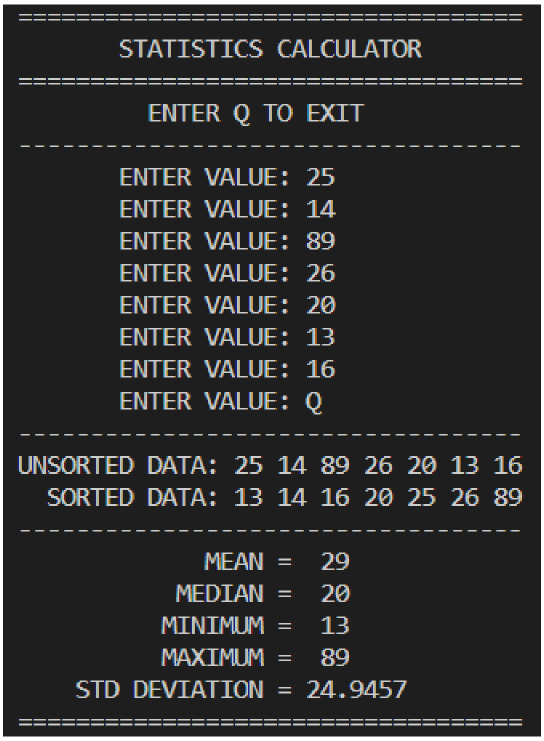
```
===================================
       STATISTICS CALCULATOR
===================================
         ENTER Q TO EXIT
-----------------------------------
       ENTER VALUE: 2
       ENTER VALUE: 3
       ENTER VALUE: 1
       ENTER VALUE: Q
-----------------------------------
UNSORTED DATA:   2  3  1
  SORTED DATA:   1  2  3
-----------------------------------
             MEAN =   2
           MEDIAN =   2
          MINIMUM =   1
          MAXIMUM =   3
    STD DEVIATION = 0.816497
===================================
```

Task (2)

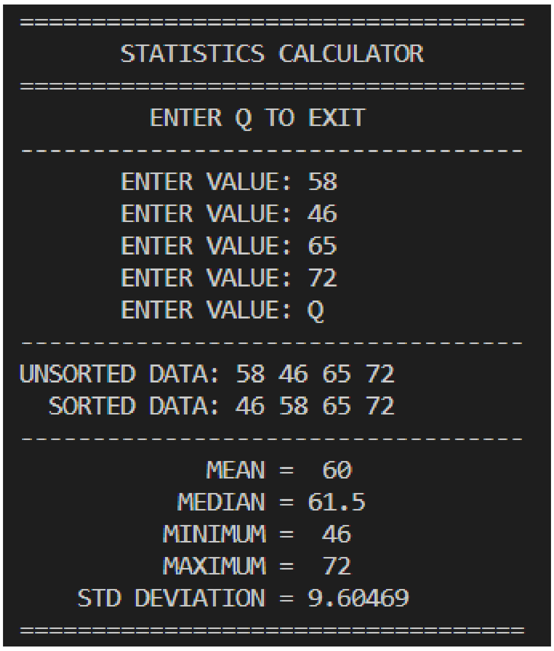
```cpp
#include<iostream>
#include<iomanip>
#include<vector>
#include<algorithm>
#include<cmath>

using namespace std;

void insertionSort(vector<int> &data)
{
    int i, hold, j;

    for ( i = 1; i < data.size(); i++)
    {
        hold = data[i];
        j = i - 1;

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

        data[j + 1] = hold;
    }

}

double mean (vector<int> &data, int sum)
{
    return (sum / data.size() );
}

double median(vector<int> &data)
{
    double mid;
    if ( data.size() % 2 != 0 )
    {
        return (double) data[ data.size() / 2];
    }
    else
    {
        return (double)( data[ ( data.size() - 1) / 2 ] + data[ data.size() / 2] ) / 2.0;
    }
}

int minimum(vector<int> &data)
{
    return data.front();
}

int maximum(vector<int> &data)
{
    return data.back();
}

double standardDeviation(vector<int> &data, int sum)
{
    double stdDvn = 0.0;

    for(int i = 0; i < data.size(); ++i)
    {
        stdDvn += pow(data[i] - mean(data, sum), 2);
    }

    return sqrt(stdDvn / data.size() );
}

int main()
{
    int value = 0;
    double sum = 0.0;
    int n = 0;
    vector<int> data;
```



```
    cout << "===================================" << endl;
    cout << "       STATISTICS CALCULATOR       " << endl;
    cout << "===================================" << endl;
    cout << "         ENTER Q TO EXIT         " << endl;
    cout << "-----------------------------------" << endl;

    cout << setw(20) << "ENTER VALUE: ";
    while (  cin >> value )
    {
        cout << setw(20) << "ENTER VALUE: ";
        data.push_back( value );
    }

    for (size_t i = 0; i < data.size(); i++)
    {
        sum += data[i];
    }

    cout << "-----------------------------------" << endl;

    cout << setw(14) << "UNSORTED DATA:";
    for (size_t i = 0; i < data.size(); i++)
    {
        cout << setw(3) << data[i];
    }
    cout << endl;

    insertionSort(data);

    cout << setw(14) << "SORTED DATA:";
    for (size_t i = 0; i < data.size(); i++)
    {
        cout << setw(3) << data[i];
    }
    cout << endl;

    cout << "-----------------------------------" << endl;

    cout << setw(20) << "MEAN = " << setw(3) <<  mean(data, sum) << endl;
    cout << setw(20) << "MEDIAN = " << setw(3) << median(data) << endl;
    cout << setw(20) << "MINIMUM = " << setw(3) << minimum(data) << endl;
    cout << setw(20) << "MAXIMUM = "  << setw(3) << maximum(data) << endl;
    cout << setw(20) << "STD DEVIATION = "  << setw(3) << standardDeviation(data, sum) << endl;

    cout << "===================================" << endl;


    return 0;
}
```



```
===================================
       STATISTICS CALCULATOR
===================================
         ENTER Q TO EXIT
-----------------------------------
       ENTER VALUE: 25
       ENTER VALUE: 14
       ENTER VALUE: 89
       ENTER VALUE: 26
       ENTER VALUE: 20
       ENTER VALUE: 13
       ENTER VALUE: 16
       ENTER VALUE: Q
-----------------------------------
UNSORTED DATA: 25 14 89 26 20 13 16
  SORTED DATA: 13 14 16 20 25 26 89
-----------------------------------
             MEAN =   29
           MEDIAN =   20
          MINIMUM =   13
          MAXIMUM =   89
    STD DEVIATION = 24.9457
===================================
```



```
===================================
       STATISTICS CALCULATOR
===================================
         ENTER Q TO EXIT
-----------------------------------
       ENTER VALUE: 58
       ENTER VALUE: 46
       ENTER VALUE: 65
       ENTER VALUE: 72
       ENTER VALUE: Q
-----------------------------------
UNSORTED DATA: 58 46 65 72
  SORTED DATA: 46 58 65 72
-----------------------------------
             MEAN =   60
           MEDIAN = 61.5
          MINIMUM =   46
          MAXIMUM =   72
    STD DEVIATION = 9.60469
===================================
```