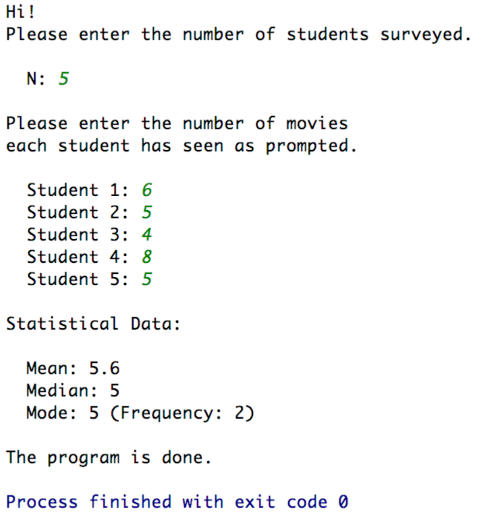
Julius Ranoa
CSC 121 001 Computer Science I
Homework – Chapter 10 Pointers

Programming Challenge @ Page 702.
Qn. 7 – Movie Statistics.

Screenshot of Runtime:



```
Hi!
Please enter the number of students surveyed.

  N: 5

Please enter the number of movies
each student has seen as prompted.

  Student 1: 6
  Student 2: 5
  Student 3: 4
  Student 4: 8
  Student 5: 5

Statistical Data:

  Mean: 5.6
  Median: 5
  Mode: 5 (Frequency: 2)

The program is done.

Process finished with exit code 0
```

Source Code:

1. *MovieStats.h*
2. *MovieStats.cpp*
3. *main.cpp*

The source code is also stored at Github.
Link below:

https://github.com/TheLoneWoof1102/FA17_CSC121001/tree/master/Source%20Code/

main.cpp

```
#include "MovieStats.h"
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    MovieStats ms;
    int count;

    cout << "Hi!" << endl;
    cout << "Please enter the number of students surveyed." << endl << endl;

    cout << "  N: ";
    cin >> count;
    ms.setNumRecords(count);

    cout << endl;
    cout << "Please enter the number of movies " << endl
         << "each student has seen as prompted." << endl << endl;

    for (int i = 0; i < count; i++) {
        int temp;
        cout << "  Student " << i + 1 << ": ";
        cin >> temp;
        ms.setRecordByIndex(i, temp);
    }

    cout << endl;
    cout << "Statistical Data: "
         << endl << endl;

    cout << "  "; ms.printMean();
    cout << "  "; ms.printMedian();
    cout << "  "; ms.printMode();

    cout << endl;
    cout << "The program is done." << endl;

    return 0;
}
```

// END of main.cpp.

MovieStats.h

```
#ifndef HOMEWORK_CH10_QN7_MOVIESTATS_H
#define HOMEWORK_CH10_QN7_MOVIESTATS_H

#include <string>

class MovieStats {
private:
    // std::string *nameList;
    int *nSeenList;
    int nRecords;

public:
    MovieStats();
    ~MovieStats();

    void setNumRecords(int);
    void setRecordByIndex(int, int);
    int getNSeenByIndex(int);
    void sortRecordsByNSeen();

    int getLength();
    double printMean();
    double printMedian();
    int printMode();
};

#endif //HOMEWORK_CH10_QN7_MOVIESTATS_H
```

// END of MovieStats.h.

MovieStats.cpp

```cpp
#include "MovieStats.h"
#include <iostream>

MovieStats::MovieStats() {
    nSeenList = nullptr;
    nRecords = false;
}

MovieStats::~MovieStats() {
    delete [] nSeenList;
    nSeenList = nullptr;
}

void MovieStats::setNumRecords(int size) {
    delete [] nSeenList;
    nSeenList = new int[size];
    nRecords = size;
}

int MovieStats::getNSeenByIndex(int idx) {
    if ( idx < 0 || idx >= nRecords ) return 0;
    else return *(nSeenList + idx);
}

void MovieStats::setRecordByIndex(int idx, int nSeen) {
    if ( idx < 0 || idx >= nRecords ) return;
    else {
        *(nSeenList + idx) = nSeen;
    }
}

void MovieStats::sortRecordsByNSeen() {
    // Selection Sort.
    int min_val, min_idx;

    for (int i = 0; i < nRecords; i++) {
        min_idx = i;
        min_val = *(nSeenList + i);
        for (int j = i + 1; j < nRecords; j++) {
            if ( *(nSeenList + j) < min_val ) {
                min_idx = j;
                min_val = *(nSeenList + j);
            }
        }
        if (min_idx != i) {
            *(nSeenList + min_idx) = *(nSeenList + i);
            *(nSeenList + i) = min_val;
        }
    }
}

int MovieStats::getLength() {
    return nRecords;
}
```

// MovieStats.cpp continued on next page.

MovieStats.cpp

```cpp
double MovieStats::printMean() {
    double total = 0.0, mean;
    for (int i = 0; i < nRecords; i++) {
        total += *(nSeenList + i);
    }
    mean = ( total / nRecords );

    std::cout << "Mean: " << mean << std::endl;
    return mean;
}

double MovieStats::printMedian() {
    sortRecordsByNSeen();

    double median = 0;
    int med_idx = (nRecords - 1) / 2;
    if (nRecords % 2 == 1) { // if odd
        median = *(nSeenList + med_idx);
    } else { // if even
        median = *(nSeenList + med_idx) + *(nSeenList + med_idx + 1);
        median /= 2;
    }

    std::cout << "Median: " << median << std::endl;
    return median;
}

int MovieStats::printMode() {
    sortRecordsByNSeen();

    int mode = *(nSeenList), modeCount = 1;
    int tempMode = mode, tempCount = 1;

    for (int i = 1; i < nRecords; i++) {
        if (tempMode == *(nSeenList + i)) {
            tempCount++;
            continue;
        } else {
            if (tempCount > modeCount) {
                mode = tempMode;
                modeCount = tempCount;
            }
            tempMode = *(nSeenList + i);
            tempCount = 1;
        }
    }

    if (tempCount > modeCount) {
        mode = tempMode;
        modeCount = tempCount;
    }

    std::cout << "Mode: " << mode
              << " (Frequency: " << modeCount << ") "
              << std::endl;
    return mode;
}
```

// END of MovieStats.cpp.