

Nick Krisulevich

Dr. Wang

COSC 120-751

11/20/2020

## Lab 7

\*\*\*\*\*

### Lab 7.1

**Source Code:** sdf

**Output:**

**Question Answer:**

Exercise 1 wanted to complete the program. Exercise 2 wanted to run the program and record the output. The output has some issues with the operations performed by the functions.

Please input a grade from 1 to 100, (or -99 to stop)

100

Enter a grade.

99

Enter a grade.

98

Enter a grade.

89

Enter a grade.

-99

The amount of grades is 4

The average of all the grades is 4.68973e+08

The lowest grade is -1875893184

```
* * * * *
```

```
* * * * *
```

### Source Code:

```
#include <iostream>

#include <iomanip>

using namespace std;

const int MAXGRADE = 25;    // maximum number of grades per student

const int MAXCHAR = 30;    // maximum characters used in a name

typedef char StringType30[MAXCHAR + 1];    // character array data type for names
                                           // having 30 characters
                                           // or less.

typedef float GradeType[MAXGRADE];    // one dimensional integer array data type

float findGradeAvg(GradeType, int);    // finds grade average by taking array of
                                       // grades and number
                                       // of grades as parameters

char findLetterGrade(float);    // finds letter grade from average given
                               // to it as a parameter
```

```

int main()
{
    StringType30 firstname, lastname;    // two arrays of characters defined
    int numOfGrades;                      // holds the number of grades
    GradeType grades;                     // grades defined as a one dimensional
array
    float average;                        // holds the average of a student's
grade
    char moreInput;                       // determines if there is more
input

    cout << setprecision(2) << fixed << showpoint;

    // Input the number of grades for each student
    cout << "Please input the number of grades each student will receive." << endl
        << "This must be a number between 1 and " << MAXGRADE << " inclusive"
        << endl;
    cin >> numOfGrades;

    while (numOfGrades > MAXGRADE || numOfGrades < 1)
    {
        cout << "Please input the number of grades for each student." << endl
            << "This must be a number between 1 and " << MAXGRADE
            << " inclusive\n";
        cin >> numOfGrades;
    }

    // Input names and grades for each student
    cout << "Please input a y if you want to input more students"
        << " any other character will stop the input" << endl;

```

```
cin >> moreInput;
```

```
while (moreInput == 'y' || moreInput == 'Y')
```

```
{
```

```
    cout << "Please input the first name of the student" << endl;
```

```
    cin >> firstname;
```

```
    cout << endl << "Please input the last name of the student" << endl;
```

```
    cin >> lastname;
```

```
    for (int count = 0; count < numOfGrades; count++)
```

```
    {
```

```
        cout << endl << "Please input a grade" << endl;
```

```
        cin >> grades[count];
```

```
    }
```

```
    cout << firstname << " " << lastname << " has an average of ";
```

```
    float finalgrade = findGradeAvg(grades, numOfGrades);
```

```
    cout << finalgrade; // Fill in code to get and print average of student to screen
```

```
    cout << " which gives them a letter grade of " << findLetterGrade(finalgrade); // Fill in call  
to get and print letter grade of student to screen
```

```
    cout << endl << endl << endl;
```

```
    cout << "Please input a y if you want to input more students"
```

```

        << " any other character will stop the input" << endl;

        cin >> moreInput;

    }

    return 0;

}

//*****

// findGradeAvg
//
// task:    This function finds the average of the
//          numbers stored in an array.
//
// data in:    an array of integer numbers
// data returned: the average of all numbers in the array
//
//*****

float findGradeAvg(GradeType array, int numGrades)
{
    // Fill in the code for this function

    int total = 0;

    for (int i = 0; i < numGrades; i++)
    {
        total += array[i];
    }

    float average = total / numGrades;

    return average;
}

```

```

//*****
// findLetterGrade
//
// task:   This function finds the letter grade for the number
//         passed to it by the calling function
//
// data in:      a floating point number
// data returned: the grade (based on a 10 point spread) based on the
//               number passed to the function
//
//*****

```

```

char findLetterGrade(float numGrade)
{
    // Fill in the code for this function
    char letter;
    if (numGrade >= 90)
    {
        letter = 'A';
    }
    else if(numGrade >= 80 && numGrade < 89.9)
    {
        letter = 'B';
    }
    else if(numGrade >= 70 && numGrade < 79.9)
    {
        letter = 'C';
    }
}

```

```
else if(numGrade >= 60 && numGrade < 69.9)
{
    letter = 'D';
}
else
{
    letter = 'F';
}
return letter;
}
```

**Output:**

Please input the number of grades each student will receive.

This must be a number between 1 and 25 inclusive

3

Please input a y if you want to input more students any other character will stop the input

y

Please input the first name of the student

mike

Please input the last name of the student

sauer

Please input a grade

100

Please input a grade

90

Please input a grade

80

mike sauer has an average of 90.00 which gives them a letter grade of A

Please input a y if you want to input more students any other character will stop the input

y

Please input the first name of the student

ron

Please input the last name of the student

cheecox

Please input a grade

88

Please input a grade

76

Please input a grade

92

ron cheecox has an average of 85.00 which gives them a letter grade of B

### Question Answer:

Exercise 1 wanted to complete and run the program

\*\*\*\*\*

\*\*\*\*\*

### Lab 7.3



### Source Code:

```
#include <iostream>

#include <iomanip>

using namespace std;

const int MAXROWS = 10;

const int MAXCOLS = 10;

typedef float PriceType[MAXROWS][MAXCOLS]; // creates a new data type
                                             // of a 2D array
                                             // of floats

void getPrices(PriceType, int&, int&);      // gets the prices into the array
void printPrices(PriceType, int, int);      // prints data as a table

int main()
{
    int rowsUsed;          // holds the number of rows used
    int colsUsed;          // holds the number of columns used
    PriceType priceTable;  // a 2D array holding the prices

    getPrices(priceTable, rowsUsed, colsUsed); // calls getPrices to fill the array
    printPrices(priceTable, rowsUsed, colsUsed); // calls printPrices to display array

    return 0;
}

//*****
//    getPrices
```

```
//
// task: This procedure asks the user to input the number of rows and
// columns. It then asks the user to input (rows * columns) number of
// prices. The data is placed in the array.
// data in: none
// data out: an array filled with numbers and the number of rows
// and columns used.
//
//*****
```

```
void getPrices(PriceType table, int& numOfRows, int& numOfCols)
{
    cout << "Please input the number of rows from 1 to " << MAXROWS << endl;
    cin >> numOfRows;

    cout << "Please input the number of columns from 1 to " << MAXCOLS << endl;
    cin >> numOfCols;

    for (int row = 0; row < numOfRows; row++)
    {
        for (int col = 0; col < numOfCols; col++)
        {
            // Fill in the code to read and store the next value in the array
            cout << "Enter a price" << endl;
            cin >> table[row][col];
        }
    }
}
```

```

//*****
//    printPrices
//
//    task:    This procedure prints the table of prices
//    data in: an array of floating point numbers and the number of rows
//             and columns used.
//    data out: none
//
//*****

```

```

void printPrices(PriceType table, int numOfRows, int numOfCols)
{
    cout << fixed << showpoint << setprecision(2);

    for (int row = 0; row < numOfRows; row++)
    {
        for (int col = 0; col < numOfCols; col++)
        {
            cout << table[row][col] << " ";
        }
        cout << endl;
    }
}

```

### Output:

Please input the number of rows from 1 to 10

2

Please input the number of columns from 1 to 10

2

Enter a price

13.37

Enter a price

12.24

Enter a price

7.77

Enter a price

5.56

13.37 12.24

7.77 5.56

### Question Answer:

Exercise 1 wanted to complete the functions.

Exercise 2 wanted to know why getPrices was passed by reference and printPrices was passed by value. getPrices was passed by reference because the function was assigning values to the array in its location in memory. Passing this function by value would have created a copy of the array in memory and would not have assigned the values to the correct location. PrintPrices is passed by value because the function takes those values from their location in memory and outputs them.

Exercise 3 wanted to know what the function findHighestPrice does. This function has two nested for loops that assign the highest value in the 2 dimensional array to highestPrice. The function then returns highestPrice.

\*\*\*\*\*

I did the assignment late Friday night and ran out of time to be able to finish the last of the exercises