COSC220 Computer Science II Review Lab

1. Write function `findAverage` to find the average value for each row in a 2D array and store them in a 1D array. The function takes three parameters: a 2D integer array (with 10 columns), the number of rows and 1D array of float to keep the average value of each row in the first 2Darray. (see below for example of what the function can generate - Array 2 is used to keep the average of each row in Array 1). Write a main function to test `findAverage`.

Array 1:

Row 1	98	90	95	100
Row 2	80	85	100	97
Row 3	60	68	80	98
Row 4	78	68	90	80
Row 5	90	95	85	83

Array 2:

95.75
90.5
76.5
79
88.25

2. Define a **Point** class. The class has two private floating point data members: `x, y`, the coordinate of a point in a two-dimensional space. The class has the following public member functions:
   - One default constructor takes no parameter and sets the value of `x, y` to 0.
   - One constructor takes two parameters and sets the value of `x, y` to the corresponding parameters.
   - A function, `setX(float xc)` sets the value of `x` to `xc`.
   - A function, `setY(float yc)` sets the value of `y` to `yc`.
   - A function, `getX()const` returns the value of `x`.
   - A function, `getY()const` returns the value of `y`.

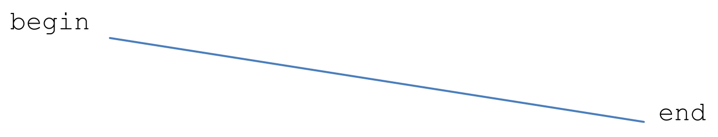
   a. Define **Point** class.
   b. Implement all the member functions**.**
   c. Write a free (or friend or member) function to overload `operator+`, which returns the distance between two **Point** objects. i.e. this following code will be allowed if `operator+` is overloaded according to such specification.

   ```
   Point p1, p2;
   int dist;

   dist = p1 + p2;
   cout << "The distance between p1 and p2 is " << dist << endl;
   ```

   d. Write a main function to test your class definition and all its member functions and the operator overloading function.

3. A line segment is represented by class **Line** given below. The two end points of a line segment are represented by the two private date members: `begin, end` which are objects of class **Point**. Class **Point** is defined in Question 7.



```
class Line
```

```
{
      public:
            // constructors
            Line();
            Line(Point b, Point e);
            // set the beginning point of the Line
            void setBegin(Point b);
            // set the end point of the Line
            void setEnd(Point e);
            // return the length of the line segment
            double getLength() const;
            // display the x and y coordinate of begin and end point.
            void showCoordinate() const;

      private: // Point is the class you defined in question 4
            Point begin;
            Point end;
};
```

a. Implement function `MakeLineArray` according to the following in-line comments.

```
// Constructs and returns an array of Line objects.
 Line * MakeLineArray(int size)
{
}
```

b. Implement function `Shortest` according to the following in-line comments.

```
// Returns an object of Line that is the shortest
// among all the Line segments associated with ptr
Line Shortest (Line * ptr, int size)
{
}
```

c. Implement function `sortLine` according to the following in-line comments.

```
// Use Bubble Sort to sort the Line objects associated with ptr.
// Sort function will sort those objects using the length in ascending order.
void sortLine (Line * ptr, int size)
{
}
```

d. Complete the following main program according to the given comments:

```
int main()
{
   Line * ptr, shortest;

   // call MakeLineArray to create an array of 100 pointers
   // to Line objects


   // For each Line object created above, ask user to enter
   // the x and y coordinate for the two end points and
   // call appropriate member functions to set the coordinate of
   // the two end points of each Line object.


   // Call appropriate function to find the shortest line segment



   // Call appropriate member functions to display the coordinate of
   // the two end points of the shortest line segment

   return 0;
}
```