CIS 330: Project #2C
Assigned: April 20th, 2016
Due April 26th, 2016
(which means submitted by 6am on April 27th, 2016)
Worth 4% of your grade

Assignment: You will implement 3 structs and 9 functions. The prototypes for the functions are located in the file prototypes.h (available on the website).

The three structs are Rectangle, Circle, and Triangle, and are described below.

The 3 structs refer to 3 different shapes: Triangle, Circle, and Rectangle.
For each shape, there are 3 functions: Initialize, GetArea, and GetBoundingBox.
You must implement 9 functions total (3*3).

The prototypes for these 9 functions are available in the file prototypes.h

There is also a driver program, and correct output for the driver program.

Again, your job is to define 3 structs and 9 functions. The comments below clarify the format of the Rectangle, Circle, and Triangle, as well as the convention for GetBoundingBox, and an example of accessing data members for pointers to structs.

== Rectangle ==

The rectangle has corners (minX, minY), (maxX, minY), (minX, maxY), (maxX, maxY).
Its area is (maxX-minX)*(maxY-minY).
Its bounding box is from minX to maxX in X, and minY to maxY in Y.

== Circle ==

The circle has an origin (x and y) and a radius.

Its area is 3.14159*radius*radius.
Its bounding box is from (x-radius) to (x+radius) in X, and (y-radius) to (y+radius) in Y.

== Triangle ==

The triangle always has two points at the minimum Y-value. The third point's Y-value is at the maximum Y-value, and its X-value is at the average of the X's of the other two points. Saying it another way, the first two points form the "base", and the third point is "height" above it.

Thus, the area of the triangle is (pt2X-pt1X)*(maxY-minY)/2;
And the bounding box is from pt1X to pt2X in X, and from minY to maxY in Y.

== GetBoundingBox ==

The GetBoundingBox calls take a double * as an argument. If a shape has its minimum X at "a", its maximum X at "b", its minimum Y at "c", and its maximum Y at "d", then it should do something like:

```
void GetCircleBoundingBox(Circle *, double *bbox)
{
   bbox[0] = a;
   bbox[1] = b;
   bbox[2] = c;
   bbox[3] = d;
}
```

== Working with pointers to structs ==

We reviewed the way to access struct data members in class, which was with the "." operator. We did not review the way to access struct data members when you have a *pointer* to a struct. And the 9 function prototypes all use pointers to structs. It is done with the ->.

So:

```
typedef struct
{
   int X;
} Y;
int main()
{
  Y y;
  Y *y2;
  y2 = &y;
  y.x = 0;
  y2->x = 1;
}
```

== What to modify ==

You will need to modify my_struct.h and my_struct.c. You should **not** modify prototypes.h or driver_2C.c. If you modify they latter two files, you will have points deducted.

== Success ==

You should run your program as:

```
./project_2C > my_output
```

and then call:

```
diff my_output driver_output
```

If diff returns no differences, then you have done the project successfully and you are ready to submit.

== What to turn in ==

Make a file called “README”
In that file, notify the reader whether you think your program is correct or not.

```
% tar cvf 2C_turnin.tar my_struct.h my_struct.c my_output README
```