Ball Collisions

PADDLE COLLISIONS:

So you can move around your paddle, and the computer moves its own paddle, but every time the ball hits either one of the paddles it just goes right through it. We need to make the ball change directions when it comes into contact with a paddle. To do this, create two separate functions, one that will handle the left paddle, and one that will handle the right paddle. In these functions:

- Make a statement that says when the Y coordinates of the ball = the Y coordinates of the paddle, change the ball direction.
    - If the ball was heading in the direction of 1, and it encountered a paddle, the ball direction should be reversed to -1, so the ball will be drawn the opposite direction.
- Doing this alone will only enable the ball to bounce back in a straight line from the direction which it came. The ball needs to be able to bounce back in different directions.
    - Segment up the paddle into different regions. When the ball contacts the middle region, it bounces off straight, if the ball contacts and upper region, it bounces off in an upward angled direction, and when the ball contacts a lower region, it bounces off in a lower angled direction.
        - Once you have this correctly functioning, you may segment the paddle into even more regions to include different angles of trajectory. You can also try to increase the variation by differentiating angles of trajectory based on not only the region of impact, but the direction of the paddle as well.

BORDER COLLISIONS:

You will also want to create two other functions that will handle collisions with the top and bottom borders. These two functions should be relatively short.

- If the ball's x coordinate is the same as the top border or bottom border, reverse the ball's direction.
    - If the ball's x coordinate = the top border coordinate, xdir = -1; If the coordinate = the bottom border, xdir = 1.

Now that you have a set of functions to handle these collision cases, call them all subsequently in your void loop and test if they work. Confirm that all function without error.

BOUNDS CASES:

The last step is to handle the out of bounds cases. You will now create one final function to take care of this. In this function:

- Check if the ball's y coordinate has gone past either the left or right paddles Y coordinate.
- If it has, reset the ball by redrawing the ball in the center of the screen, and resetting it's Y/X coordinates accordingly

Now call this function in your void loop as well. Test and confirm that all functions are fully operational.