A7: Mirroring part of a picture

Sometimes you only want to mirror part of a picture. For example, Figure 10 shows a temple in Greece that is missing a part of the roof called the pediment. You can use the explorer tool to find the area that you want to mirror to produce the picture on the right. If you do this you will find that you can mirror the rows from 27 to 96 (inclusive) and the columns from 13 to 275 (inclusive). You can change the starting and ending points for the row and column values to mirror just part of the picture.

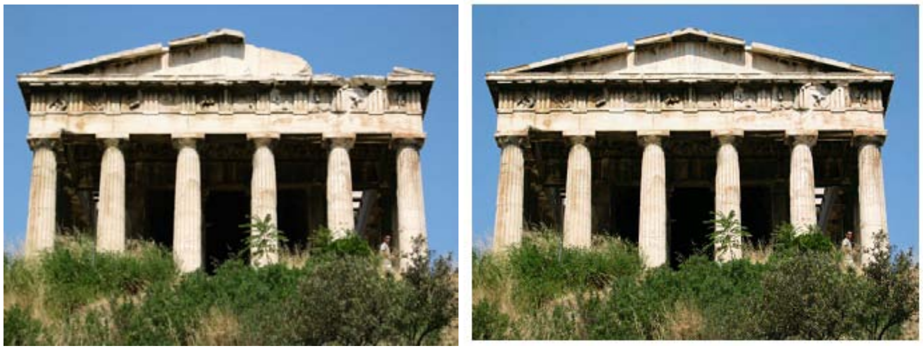

Figure 10: Greek temple before (left) and after (right) mirroring the pediment

To work with just part of a picture, change the starting and ending values for the nested `for` loops as shown in the following `mirrorTemple` method. This method also calculates the distance the current column is from the `mirrorPoint` and then adds that distance to the `mirrorPoint` to get the column to copy to.

```
  public void mirrorTemple()
  {
    int mirrorPoint = 276;
    Pixel leftPixel = null;
    Pixel rightPixel = null;
    int count = 0;
    Pixel[][] pixels = this.getPixels2D();

    // loop through the rows
    for (int row = 27; row < 97; row++)
    {
      // loop from 13 to just before the mirror point
      for (int col = 13; col < mirrorPoint; col++)
      {

        leftPixel = pixels[row][col];
        rightPixel = pixels[row]
                         [mirrorPoint - col + mirrorPoint];
        rightPixel.setColor(leftPixel.getColor());
      }
      }
    }
```

You can test this with the `testMirrorTemple` method in `PictureTester`.

How many times was `leftPixel = pixels[row][col];` executed? The formula for the number of times a nested loop executes is the number of times the *outer loop* executes multiplied by the number of times the *inner loop* executes. The outer loop is the one looping through the rows, because it is outside the other loop. The inner loop is the one looping through the columns, because it is inside the row loop.

How many times does the outer loop execute? The outer loop starts with `row` equal to 27 and ends when it reaches 97, so the last time through the loop `row` is 96. To calculate the number of times a loop executes, subtract the starting value from the ending value and add one. The outer loop executes 96 – 27 + 1 times, which equals 70 times. The inner loop starts with `col` equal to 13 and ends when it reaches 276, so, the last time through the loop, `col` will be 275. It executes 275 – 13 + 1 times, which equals 263 times. The total is 70 * 263, which equals 18,410.

Questions

1. How many times would the body of this nested `for` loop execute?
   ```
   for (int row = 7; row < 17; row++)
     for (int col = 6; col < 15; col++)
   ```
2. How many times would the body of this nested `for` loop execute?
   ```
   for (int row = 5; row <= 11; row++)
     for (int col = 3; col <= 18; col++)
   ```

Exercises

1. Check the calculation of the number of times the body of the nested loop executes by adding an integer `count` variable to the `mirrorTemple` method that starts out at 0 and increments inside the body of the loop. Print the value of `count` after the nested loop ends.
2. Write the method `mirrorArms` to mirror the arms on the snowman (“`snowman.jpg`”) to make a snowman with 4 arms. Write a class (static) test method in `PictureTester` to test this new method and call it in the `main` method.
3. Write the method `mirrorGull` to mirror the seagull (“`seagull.jpg`”) to the right so that there are two seagulls on the beach near each other. Write a class (static) test method in `PictureTester` to test this new method and call it in the `main` method.