Section 10: While Loops

Part 1: While Loops Mystery

Consider the following method. For each call listed below, give what output is produced.

```
public static int mystery(int num) {
      int result = 0;
      while (num > 0) {
            result = result * 10 + num % 10;
            num = num / 10;
      }

      return result;
}
```

`mystery(5);`

```
5
```

`mystery(52);`

```
25
```

`mystery(108);`

```
801
```

Part 2:

You are writing a method called `luckyNumberGame` that simulates a number guessing game with the following rules:

- Start with a target number (the "lucky number")
- Keep generating random numbers between 1-20 (inclusive)
- For each number generated:
    - If it's even, add it to your score
    - If it's odd and greater than 10, subtract 5 from your score
    - If your score ever goes negative, reset it to 0
- Stop when you generate the lucky number
- Print the total numbers generated and the final score

For example, `luckyNumberGame(rand, 7)` with random sequence `[4, 15, 12, 7]` should print:

```
Generated 4, score is now: 4
Generated 15, score is now: -1
Score reset to 0!
Generated 12, score is now: 12
Generated 7, that's the lucky number!
Total numbers generated: 4
Final score: 12
```

Here is a buggy implementation of `luckyNumberGame()`:

```
public static void luckyNumberGame(Random rand, int luckyNum) {
      int score = 0;
      int count = 0;
      int num = rand.nextInt(20) + 1;
      while (num != luckyNum) {
            count++;
            System.out.println("Generated " + num +
                  ", score is now: " + score);

            if (num % 2 == 0) {
                  score += num;
            } else if (num > 10) {
                  score -= 5;
            }

            if (score < 0) {
                  score = 0;
                  System.out.println("Score reset to 0!");
            }

            num = rand.nextInt(20) + 1;
      }

      System.out.println("Generated " + num +
            ", that's the lucky number!");
      System.out.println("Total numbers generated: " + count);
      System.out.println("Final score: " + score);
}
```

Step 1: Trace

Assume the `Random` object generates the sequence `[4, 15, 12, 7]`
Trace through `luckyNumberGame(rand, 7)` with the buggy code. Show all variable values after each iteration.

Iteration	`num`	`count`	`score` (printed)	`score` (after update)	`num` (next)
Before Loop	4	0	----------	0	4
1	4	1	0	4	15
2	15	2	4	0 (reset)	12
3	12	2	0	12	7
After Loop	7	3	12	----------	----------

What output does the buggy code produce?

```
Generated 4, score is now: 0
Generated 15, score is now: 4
Score reset to 0!
Generated 12, score is now: 0
Generated 7, that's the lucky number!
Total numbers generated: 3
Final score: 12
```

Step 2: Identify Bugs

Describe all the bugs you found in the code:

1. **The score is printed BEFORE being updated.** The print statement appears before the score calculation, so it always displays the previous score value instead of the current one. For example, when we generate 4, we print "score is now 0" before adding 4 to it.
2. **The count doesn't include the lucky number.** We only increment count inside the while loop, which stops before processing the lucky number. The total should be 4 (including the lucky number 7), but we only count 3.

Part 3: Fix

Rewrite the method to produce the correct output:

```
public static void luckyNumberGame(Random rand, int luckyNum) {
	int score = 0;
	int count = 0;
	int num = rand.nextInt(20) + 1;
	while (num != luckyNum) {
		count++;

		// Update score FIRST
		if (num % 2 == 0) {
			score += num;
		} else if (num > 10) {
			score -= 5;
		}

		// THEN print the updated score
		System.out.println("Generated " + num +
			", score is now: " + score);


		if (score < 0) {
			score = 0;
			System.out.println("Score reset to 0!");
		}

		num = rand.nextInt(20) + 1;
	}

	// Include the lucky number in the count
	count++;

	System.out.println("Generated " + num +
		", that's the lucky number!");
	System.out.println("Total numbers generated: " + count);
	System.out.println("Final score: " + score);
}
```