Practice 7: Simple Counters

Using iterators and defined loops (equivalent to the *for* loop)
Module 2: Flow Control

Don't repeat it yourself, let the computer do it
One of the greatest advantages of computing is the ability to repeat a task thousands of times without getting tired or making mistakes. In programming, this is called **iteration** or a **loop**. In this practice, we will learn to use defined loops, where we know exactly how many times we want something to repeat.

Practice Objectives

- Understand the concept of a **defined loop**.
- Learn to use a **control variable** (index) that changes automatically.
- Automate numerical sequences (counting, multiplication tables, etc.).

Key Concepts: The "For each" Loop

In Snap!, the most powerful block for counting is the orange block in the Control category: `for (i) = (1) to (10)`.

- **The variable (i):** This is a variable that the loop creates for us. It starts with the first number and increases by 1 until it reaches the second number.
- **The loop body:** Everything you put inside the block will be executed at each step.

THE CHALLENGE: Countdown and the Table

Part A: Rocket Launch
Create a program that performs a countdown from 10 to 1.

1. Use the block `for (i) = (1) to (10)`.
2. **Tip:** To count backwards, the sprite should say the result of the operation: `(11 - i)`.
3. At the end of the loop, add a block to say `"LIFT OFF!"`.

Part B: Multiplication Tables
Ask the user for a number and show its multiplication table from 1 to 10.

1. Ask: `"Which table do you want to see?"` and save the `answer` in a variable called `table_of`.

2. Use the block `for (i) = (1) to (10)`.
3. Inside, use the `say` block combined with the green `join` block to show something like: *"5 x 3 = 15"*.

Solution Pseudocode (Multiplication Table)

In pseudocode, a defined loop is written using the `For ... To ... Do` structure.

```
Algorithm MultiplicationTable
   Define n, i, result As Integer
   Write "Which table do you want to calculate?"
   Read n

   For i <- 1 To 10 Step 1 Do
      result <- n * i
      Write n, " x ", i, " = ", result
   EndFor
EndAlgorithm
```

Note for the future

In text-based languages like C++ or Java, this loop is called **for**. Its structure is very famous: `for (int i = 1; i <= 10; i++) { ... }`

It means exactly the same: start at 1, go up to 10, and add 1 in each round.

Try changing the loop limit to 100 and see how fast your computer counts!