Day 1 - Variable Exercise

Rust Variables Practice Questions

Question 1: Basic Variables

1. Declare an immutable variable `x` with the value 5 and print it.
2. Try to change `x` to 10 - what happens? Comment out this line after you see the error.
3. Declare a mutable variable `y` with the value 3 and print it.
4. Change `y` to 7 and print it again.
5. Declare a new immutable variable `z` that contains the sum of `x` and `y`, then print it.

Question 2: Variable Shadowing

1. Declare a variable `number` with value 5 and print it.
2. Shadow the original `number` by multiplying it by 2 and print the new value.
3. Shadow `number` again by changing it to a string "five" and print it.
4. What happens if you try to use the string `number` in a mathematical operation like `number + 10`?

Question 3: Constants

1. Declare a constant `MAX_POINTS` at the global level (outside main) with value 100,000 (using underscore separator).
2. Print this constant inside main().

3. Try to change the constant's value - what happens?
4. Create another constant `MIN_POINTS` inside main() with value 10 and print it.
5. Can you shadow the global `MAX_POINTS` constant with a let binding inside main()?

Question 4: Type Annotations

1. Declare variables with these explicit types and values:
    - a : i32 with value 42
    - b : f64 with value 3.14
    - c : bool with value true
    - d : char with value 'R'

Print all four variables.

1. What happens if you try to assign 3.14 to an i32 variable?
2. What happens when you assign 42 to an f64 variable?

Question 5: Scope and Shadowing

1. Declare a variable `outer` with value 10 and print it.
2. Create a new block (with curly braces).
    - Inside, declare a variable `inner` with value 5 and print it.
    - Shadow `outer` by redeclaring it as a string "ten" and print it.
3. After the block, print `outer` again - what value does it have?
4. Try to access `inner` outside the block - what happens?

Bonus Challenge

Create a program that:

1. Starts with an immutable variable `count` set to 0
2. Uses shadowing to increment it three times (to 3)

3. Then declares it as mutable and increments it two more times (to 5)
4. Finally shadows it one last time as a string "five" and prints it

Remember to test each question by:

1. Writing the code
2. Predicting what will happen
3. Running it to see if you were right
4. Reading any error messages carefully if your code doesn't compile