Student Name: _____ Student ID: _____ TA: _____

Rust Lab 02

Date: 9/7/2025

Lab 1: Name Formatting with Loops

Write a Rust program that:
1) Prompts the user to enter two names: one for Player 1 and one for Player 2.
2) Prints the names in two different formatted patterns: one vertical and one horizontal, using loops to print the border of asterisks.

Example:
If the user inputs the names "Mike" for Player 1 and "Leo" for Player 2, the output should be:

```
Vertical Pattern:
******************
*                *
* Player 1: Mike *
*                *
******************
*                *
* Player 2: Leo  *
*                *
******************
```

```
 Horizontal Pattern:
**********************************
*                *                *
* Player 1: Mike * Player 2: Leo *
*                *                *
**********************************
```

Hint:
Read string: io::stdin().read_line(&mut player1).expect("Failed to read");
Length of string: player.len()

TA Comment: _____
TA Check for using loop, no hard code.

Lab 2: Employee Data System

Write a Rust program that simulates a simple employee data processing system:
1) Prompts the user to enter the salaries of 5 employees and stores them in an array.
2) Prompts the user to enter employee details as a tuple consisting of the employee's name, age, and salary.
3) Prints all employee details in a formatted message. Name: <name> Age: <age> Salary: <salary>
4) Prints the total salary, average salary, employee with the highest salary, and employee with the oldest age.

Example Output:
Employee 1: Name = "Alice", Age = 30, Salary = 30000
Employee 2: Name = "Bob", Age = 41, Salary = 45000
Employee 3: Name = "Charlie", Age = 38, Salary = 50000
Employee 4: Name = "Diana", Age = 43, Salary = 35000
The employee with the highest salary is: Charlie with a salary of 50000
The oldest employee is: Diana, 43 years old

TA Comment: _____ (Check for using loop, no hard code. Using array and tuple.)

Student Name: _____ Student ID: _____ TA: _____

Lab 3: Number Guessing Game with Control Flow

Write a Rust program that:

1) Uses a fixed secret number (e.g., 42).
2) Prompts the user to guess the number repeatedly.
3) Uses a loop to keep asking until the correct number is guessed.
4) On each guess, use match or if to:
   - Print "Too low" if the guess is less than the secret.
   - Print "Too high" if the guess is greater than the secret.
   - Print "Correct!" and break the loop when matched.
5) Track how many attempts were made and print the total at the end.

Sample Output:

```
Guess the number (1-100):
> 30
Too low.

Guess the number (1-100):
> 70
Too high.

Guess the number (1-100):
> 42
Correct! You guessed it in 3 tries.
```

Hint:

- Use loop and break
- Use .trim() and .parse() for input conversion
- Use match or if-else for decision logic

TA Comment: _____

TA Check for loop, match/if, input parsing, attempt tracking.