RUST Launch Pad

Worksheet # 1

1. Write a Rust program that prints "Hello, Rust!" to the console.
2. Create a Rust program that asks the user for their name and then prints a personalized greeting.
3. Write a Rust program that converts temperatures between Celsius and Fahrenheit. Allow the user to input the temperature and the scale (C or F) and print the converted temperature.
4. Implement a Rust function that calculates the factorial of a given non-negative integer using recursion.
5. Write a Rust program that checks if a given integer is even or odd and prints the result.
6. Create a Rust program that calculates the sum of all integers from 1 to N, where N is provided by the user.
7. Write a Rust program that prints numbers from 1 to 100. For multiples of 3, print "Fizz" instead of the number. For multiples of 5, print "Buzz" instead of the number. For numbers which are multiples of both 3 and 5, print "FizzBuzz."
8. Implement a Rust function that checks if a given string is a palindrome (reads the same forwards and backwards).
9. Create a Rust program that generates a random number between 1 and 100 and allows the user to guess it. Provide feedback on whether the guess is too high or too low.
10. Develop a Rust program that acts as a basic calculator. It should allow the user to enter two numbers and an operation (+, -, *, /) and display the result.
11. Write a Rust function that finds and returns the largest number in an array of integers.
12. Implement a Rust function that takes a string as input and returns the string reversed.
13. Write a Rust function that generates the Fibonacci sequence up to a specified number of terms, N.
14. Create a Rust program that generates and prints all prime numbers up to a given limit, N.
15. Develop a Rust program that demonstrates ownership and borrowing concepts. Define a function that takes ownership of a string, modifies it, and then borrows it back for printing.
16. Define a Rust function that takes two string references as parameters and returns the longest one. Include lifetime annotations in the function signature.
17. Create a Rust program that takes a string from the user and prints its first and last character using string slicing.

18. Implement a Rust function that takes a string and a substring as input and returns the number of times the substring appears in the string.
19. Write a Rust function that takes a sentence as input and reverses the order of words while preserving the order of characters within each word.
20. Create a Rust program that reads a text file and counts the occurrences of each word, ignoring punctuation and case. Use string slicing to split the text into words.