1. **Explain String Creation**:
    - Write the two ways to create a string in Java and explain the difference between them (using literals vs. `new` keyword).
2. **String Constant Pool**:
    - What is the String Constant Pool? Explain its role and why it is important for memory optimization in Java.
3. **Demonstrate Immutability**:
    - Write a program to show the immutability of strings. Create a string, modify it, and observe if the original object changes.
4. **Compare String, StringBuilder, and StringBuffer**:
    - Explain the difference between `String`, `StringBuilder`, and `StringBuffer`. Provide examples showing when each should be used.
5. **Perform Concatenation**:
    - Write a program to concatenate two strings using:
        - `+` operator.
        - `concat()` method.
        - `StringBuilder` or `StringBuffer`.
6. **Explain UTF-16 Encoding**:
    - Discuss how strings in Java use UTF-16 encoding and why it is relevant for global character support.
7. **String Manipulation**:
    - Write a program to perform the following string operations:
        - Convert to uppercase and lowercase.
        - Find the length of a string.
        - Extract a substring.
8. **Discuss the Memory Usage**:
    - Why is it better to use string literals for frequently used strings instead of creating them with the `new` keyword?
9. **Practice with Methods**:
    - Write a program using the `String` methods like `indexOf()`, `charAt()`, and `replace()`. Explain their usage.
10. **Use StringBuilder Efficiently**:

- Write a program to build a large string by appending multiple parts using `StringBuilder`. Compare the efficiency with using concatenation with `+`.