Excercise#1 (Discussion)

The following loop is intended to double the *String s* by concatenating a copy of s at the end of s (one character at a time). For example, if s initially contains JAVA , then after the loop s should contain JAVAJAVA .

for (int pos=0; pos<s.length(); pos++)
s = s + s.charAt(pos);

Is there a problem with the above code? What does the loop actually do? How could you problem?

The code enters endless loop. The reason is that every time we add letter to initial string, which means the size of that string increases in every iteration and that length is passed by reference, so it affects for loop iteration. To solve this issue, we can set the length of initial string to new variable before loop and use it in the statement of for loop.

```java
public class Main {
    public static void main(String[] args) {
        String s = "JAVA";
        int length = s.length();

        for (int pos=0; pos<length; pos++)
            s = s + s.charAt(pos);

        System.out.println(s);
    }
}
```

Excercise#2 (Coding Practice)

Write a Java program to generate a truth table. Using a command-line argument, input a positive decimal integer n. Generate the truth table that consists of 2^n entries and output the result on the console. Name your class <TruthTable>. **Your class should consist of at least two methods.**

➢   **Write a tester program to check your work prompting user to enter a positive integer number and output as shown.**

➢   **Provide sufficient comments in your code and provide appropriate input to the Java documentation**

You need to submit <u>both</u> your code and this PDF after adding your answer to Exercise#1.

For example:

PROMPT> java TruthTable 3

000
001
010
011
100
101
110
111