CSE 1322L - Lab 7

Introduction

In this lab, you will practice writing recursive methods for solving the following problems:

- Multiplication
- Division
- Remainder
- Repeating a string N times
- Check if a string is the reverse of another

Requirements

The features described below must be in your program:

- The following static methods need to be present in your driver class. Note that their **solution must be recursive; non-recursive solutions will be heavily penalized**.
    - **int recursiveMultiply(int, int):** Returns the product of the first argument by the second argument.
        - No need to worry about negative arguments
        - Note that no multiplication operation is necessary (i.e.: this method does not need the * operator)
    - **int recursiveDivision(int, int):** Returns the quotient (i.e.: the whole part of the division) of the first argument by the second argument.
        - **If the second argument is 0, the method should return -1**
        - No need to worry about negative arguments
        - Note that no division operation is necessary (i.e.: this method does not need the / operator)
    - **int recursiveRemainder(int, int):** Returns the remainder of dividing the first argument by the second argument.
        - **If the second argument is 0, the method should return -1**
        - No need to worry about negative arguments
        - Note that no remainder operation is necessary (i.e.: this method does not need the % operator)
    - **String recursiveEcho(String, int)**: Returns the first string concatenated into itself N times, where N is the second argument.
        - No need to worry about the second argument being negative
        - Note that no loops are necessary (i.e.: this method does not need a FOR, WHILE, or DO-WHILE loop)
    - **boolean recursiveReverse(String, String)**: returns true if the second string is the reverse of the first string, and false otherwise

        - **Ignore casing**
        - Note that no loops are necessary (i.e.: this method does not need a FOR, WHILE, or DO-WHILE loop)
- Main(): implement the following menu options:
    1. **Multiply 2 numbers**: Prompt the user for two numbers, then use them as arguments to call recursiveMultiply(), printing the result.
    2. **Divide 2 numbers**: Prompt the user for two numbers, then use them as arguments to call recursiveDivision(), printing the result.
    3. **Mod 2 numbers**: Prompt the user for two numbers, then use them as arguments to call recursiveRemainder(), printing the result.
    4. **Echo sentence**: Prompt the user for a sentence, and how many times it should be repeated. Pass both inputs to recursiveEcho() and print its result.
    5. **Determine if reverse**: Prompt the user for 2 sentences. Print an appropriate message depending on if the two sentences are the opposite of each other.
    6. **Quit**: Terminates the program

Deliverables

- Lab7.java

Considerations

- Recall that recursive solutions have two components to it: a base case and a recursive case.
- There is no upper limit to the number of recursive cases or base cases that your solution can have.
- This was mentioned above but it bears repeating: besides your main(), none of your other methods should have any loops in them. Your solutions need to be strictly recursive.
- Similarly, for the methods that perform mathematical operations, you do not need to use the *, /, or % operators.

Sample Output (user input in red)

```
1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 1
Enter the first number: 2
Enter the second number: 8
```

```
Your product is 16

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 1
Enter the first number: 3
Enter the second number: 12
Your product is 36

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 2
Enter the first number: 14
Enter the second number: 5
Your quotient is 2

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 2
Enter the first number: 21
Enter the second number: 3
Your quotient is 7

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 2
```

```
Enter the first number: 3
Enter the second number: 12
Your quotient is 0

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 3
Enter the first number: 14
Enter the second number: 5
Your modulus is 4

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 3
Enter the first number: 21
Enter the second number: 3
Your modulus is 0

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 3
Enter the first number: 3
Enter the second number: 12
Your modulus is 3

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
```

```
6. Quit
Enter option: 4
Enter your sentence: bake the cake!
Repeat how many times? 4
Your sentence repeated 4 times is
bake the cake!bake the cake!bake the cake!bake the cake!

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 5
Enter a sentence: Alice
Enter another sentence: Secila
The sentences are NOT the opposite of each other.

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 5
Enter a sentence: Robert
Enter another sentence: Trebor
The sentences are the opposite of each other.

1. Multiply 2 numbers
2. Divide 2 numbers
3. Mod 2 numbers
4. Echo sentence
5. Determine if reverse
6. Quit
Enter option: 6
Shutting off...
```