CSE1322L - Lab 8

Objectives

Recursion

Background

In this lab you'll write a number of recursive methods.

Lab 8A: You will write multiplication, division and modular division using recursion.
Lab 8B: Write a recursive method repeatNTimes and isReverse

Lab 8A Tasks

You'll need to write 3 methods, all of which must use recursion:

1) Write recursive_multiply. It should take in 2 integers and return an integer.

   Another way to say multiply 5*4, is 5+(5*3), or 5+5+(5*2), or 5+5+5+(5*1), or
   5+5+5+5+(5*0). You know that any number multiplied by 0 is 0.

2) Write recursive_div. It should take in 2 integers and return an integer. Unlike regular
   division, div just returns the integer portion of division. It answers the question "how
   many times does the second number go into the first number". So, 7 div 3 = 2, because 3
   goes in 2 times, the leftover is irrelevant to div.

   If you are asked to divide anything by 0, you should return -1, as that's an error.

   If you are asked to divide anything by itself, you should return 1. i.e. 7 div 7 = 1

   If you are asked to divide a small number by a bigger one, the answer is 0. i.e. 2 div 7 = 0

   For all other numbers, you'll keep subtracting the second number from the first number,
   until the first number is less than the second. You'll count how many times this happens,
   and return that count. Remember you **must use recursion in all 3 of these methods**.

3) Write recursive_mod. It should take in 2 integers and return an integer. Mod only cares
   about the remainder of division.

   If you are asked to divide anything by 0, it should return -1, as that's an error.

   If you are asked to divide any smaller number by a larger number, you should return the
   smaller number. i.e. 2 mod 3 = 2

   For all other cases, you'll keep subtracting the second number from the first.

   For example: 7 mod 3:

   7-3 = 4

4-3 = 1
Answer is 1.

4) Finally, write the main method, which prompts the user to either multiply, div, or mod. Then prompt the user for 2 numbers, call the appropriate method, receive a result, and print it out.

Continue to ask the user (you may use a loop here) until they choose 0 to quit.

Lab 8A Sample Output:
```
Choose from the following:
0.   Quit
1.   Multiply 2 numbers
2.   Div 2 numbers
3.   Mod 2 numbers
1
Enter first number
3
Enter second number
5
Answer: 15
Choose from the following:
0.   Quit
1.   Multiply 2 numbers
2.   Div 2 numbers
3.   Mod 2 numbers
1
Enter first number
5
Enter second number
0
Answer: 0
Choose from the following:
0.   Quit
1.   Multiply 2 numbers
2.   Div 2 numbers
3.   Mod 2 numbers
2
Enter first number
15
Enter second number
5
Answer: 3
```

```
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
2
Enter first number
15
Enter second number
2
Answer: 7
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
2
Enter first number
15
Enter second number
0
Answer: -1
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
2
Enter first number
2
Enter second number
10
Answer: 0
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
3
Enter first number
```

```
7
Enter second number
2
Answer: 1
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
3
Enter first number
10
Enter second number
3
Answer: 1
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
3
Enter first number
10
Enter second number
0
Answer: -1
Choose from the following:
0.  Quit
1.  Multiply 2 numbers
2.  Div 2 numbers
3.  Mod 2 numbers
0
```

Lab 8B Tasks:

- Recursive method repeatNTimes
  - Takes in a string and an integer as parameters
  - Concatenates the string to itself the number of times indicated. Returns the resulting string.
  - For example, if passed "Hi", 4. It would return "HiHiHiHi". If passed "This is a test ",2, it would return "This is a test This is a test".
  - If passed "Test",0, it should return an empty string """

- Recursive method isReverse
  - Takes in 2 strings as parameters
  - Returns a boolean indicating if the 2 strings have the same characters but reversed from eachother.
  - For example, if passed "Hello" and "olleH" it would return true
  - If passed "a" and "a" it would return true
  - If passed "Test" and "Test" it would return false.
  - 2 empty strings should be considered the reverse of each other.
  - Capitalization must also match. I.e. abcd and DCBA should return false.
  - You will may find the following methods helpful:
    - Java: string.length(), string.charAt(0), string.substring()
    - C#: string.Length, string[0], string.Substring()

- Main method:
  - Using the repeatNTimes method print out the string "I must study recursion until it makes sense\n" 100 times.
  - Next prompt the user for string 1 (e.g. s1), then prompt the user for string 2 (e.g. s2).
  - Call your isReverse method passing it the first and second strings entered by the user.
  - If they are the reverse print s1+" is the reverse of "+s2
  - If they the second is not the reverse of the first, print s1+" is not the reverse of "+s2

Lab 8B Sample Output:

```
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
```

```
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
```

```
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense
I must study recursion until it makes sense

Enter the first string
abcd
Enter the second string
dcba
abcd is the reverse of dcba

**Separate run, ignoring the first 100 lines…**
Enter the first string
a
Enter the second string
a
a is the reverse of a

**Separate run, ignoring the first 100 lines…**
Enter the first string
abcd
Enter the second string
ddba
abcd is not the reverse of ddba
```

```
**Separate run, ignoring the first 100 lines.  Note here we are
hitting return for each string, which sends empty string**
Enter the first string

Enter the second string

 is the reverse of
```

Submission Guidelines:

You will turn in two programs Lab8a and Labb.

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:
https://ccse.kennesaw.edu/fye/courseschedules.php