

German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Wael Abouelsaadat
Dr. Mohammed Abdel Megeed

Introduction to Computer Programming, Spring Term 2017
Practice Assignment 5

Discussion: 18.3.2017 - 23.3.2017

Exercise 5-1 Blast Off

Write a recursive method `countdown` that takes a single integer as a parameter from the user and prints the numbers from `n` until 1 and then prints "Blastoff!". If the parameter is zero, it prints only the word "Blastoff!"

For example if the user will enter 6 then the program should output:

```
6
5
4
3
2
1
Blastoff!
```

Exercise 5-2 Power

Consider the evaluation of x^n , where n is a non-negative integer. Write a recursive method `powerRec` to calculate x^n .

Write a `main` method to test your method.

Exercise 5-3 Natural Logarithm

Write a recursive method `constantRec` to calculate the value of the mathematical constant e which is defined as:

$$e(n) = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

Implement first a recursive method `factorial`, where $factorial(n) = n!$.

Write a `main` method to test your method.

Exercise 5-4 MultiplyRec

To be discussed in the Lab

Write a recursive method `multiplyRec` that perform the multiplication of two numbers.

$$f(x, y) = x * y$$

Your method will use only the addition and subtraction operators.

Write a `main` method to test your method.

Hint:

$$x * y = \underbrace{x + x + \dots + x}_{y \text{ times}}$$

Exercise 5-5 Division

Write a recursive Java method `divideRec` that perform the integer division operation of two numbers. If y is a negative number, an error message has to be displayed.

$$f(x, y) = \frac{x}{y}$$

DO NOT USE THE JAVA PREDEFINED OPERATOR `/`.

Hint:

$$\frac{x}{y} = \begin{cases} 0 & : \text{ if } x < y \\ 1 + \frac{x-y}{y} & : \text{ Otherwise} \end{cases}$$

Write a `main` method to test your method.

Exercise 5-6 Modulus

Write a recursive Java method `ModulusRec` that perform the modulus operation of two integers. If y is a negative number, an error message has to be displayed.

$$f(x, y) = x \% y$$

DO NOT USE THE JAVA PREDEFINED OPERATOR `%`.

Hint:

$$x \% y = \begin{cases} 0 & : \text{ if } x = 0 \\ x & : \text{ if } x < y \\ (x - y) \% y & : \text{ Otherwise} \end{cases}$$

Write a `main` method to test your method.

Exercise 5-7 Sum of Digits

Write a recursive method to determine the sum of the digits of an integer. For example, the sum of digits of 51624 is $5 + 1 + 6 + 2 + 4 = 18$.

Exercise 5-8 Number of Digits

Write a recursive Java method `numberDigitsRec`, which given an integer, returns its number of digits. For example the call `numberDigitsRec(12312)` will return 5.

Exercise 5-9 Prime

A prime number is an integer that cannot be divided by any integer other than one and itself. For example, 7 is prime because its only divisors are 1 and 7. The integer 8 is not prime because its divisors are 1, 2, 4, and 8.

Another way to define prime is:

```

prime(N)    = prime(N, N-1)

prime(N, 1) = true

prime(N, D) = false      if D divides N
              prime(N, D-1) otherwise

```

For example,

```

prime(4)    = prime(4,3)
prime(4,3)  = prime(4,2)
prime(4,2)  = false

```

Another example,

```

prime(7)    = prime(7,6)
prime(7,6)  = prime(7,5)
prime(7,5)  = prime(7,4)
prime(7,4)  = prime(7,3)
prime(7,3)  = prime(7,2)
prime(7,2)  = prime(7,1)
prime(7,1)  = true

```

Translate the math-like definition of **prime** into two Java methods that return boolean. Use the **%** operator to test divisibility. Put your method into a class, write a testing class, and test your program. (Look at `Triangle.java` in this assignment.)

Exercise 5-10 Cube numbers

Write a program that implements this definition of cube numbers, where **N** is an integer entered by the user.

```

cube(1) = 1
cube(N) = cube(N-1) + 3(square(N)) - 3N + 1

```

Implement the **square()** method using this definition (also given in Lab assignment 7):

$$(N - 1)^2 = N^2 - 2N + 1$$

Exercise 5-11 Binomial Coefficient

The binomial coefficient $\binom{n}{k}$ is the number of ways of picking k unordered outcomes from n possibilities, also known as a combination or combinatorial number.

The binomial coefficient is defined recursively as follows:

$$\binom{n}{k} = \begin{cases} 1 & : \text{ if } k = 0 \\ 1 & : \text{ if } n = k \\ \binom{n-1}{k} + \binom{n-1}{k-1} & : \text{ otherwise} \end{cases}$$

Write a recursive method to calculate the binomial coefficient. Make sure that n is less than k .

Write a **main** method that will allow the user to enter the actual parameters of the binomial coefficient method. Use either **Scanner** class or the command line arguments as input mechanism.

```

public static void main(String [] args) {
    Scanner sc = new Scanner(System.in);
    ....
}

```

Exercise 5-12 CountRec

Write a recursive method named `countRec` that accepts two arguments: a `String` value, and a `char` value. Your method is to return the total number of times the character appears inside of the string.

Exercise 5-13 Reverse

Consider reversing the characters in a string. Write a recursive Java method `reverseRec` that returns a new string with the same characters of the original string, but in reversed order.

Think about the base case and recursive case:

- Base case: `ReverseRec("") => ""`
- Recursive case:

`reverse("ABCDE") => "E" + ReverseRec("ABCD") => => "EDCBA"`

Hint: In addition to the methods `charAt` and `length`, use the predefined method `substring()`. For example if we have a string `String s = "CSEN202"`, then `s.substring(1)` returns `"SEN202"`, i.e. the string `s` without the first character.

Write a `main` method to test your method.

Exercise 5-14 Palindrome To be discussed in the Tutorial

Write a recursive method `palindrome` that takes a single string as a parameter from the user and returns whether the string is palindrome or not.

Exercise 5-15 Replace

Write a recursive method `replace` that takes two arguments: `String` and `char` and replaces each occurrence of the character with a '*'

Exercise 5-16 Eliminate

Write a recursive method `eliminate` that takes a `String` and a `char` and deletes each occurrence of this character from the string.

Exercise 5-17 Recursion Tracing 1

Give the following program:

```
public static void mystery1(int a, int b){
    if (a <= b){
        int m =(a + b) / 2;
        System.out.print(m + "_");
        mystery1(a, m - 1);
        mystery1(m+1, b);
    }
}
```

What is the output of the main method below? Justify your answer with a tracing table.

```
public static void main (String [] args){
    int n = 6;
    mystery1(0, n);
}
```

Exercise 5-18 Recursion Tracing 2
To be discussed in the tutorials

Give the following program:

```
public static void mystery(String prefix, String remaining, int k){
    if(k == 0){
        System.out.println(prefix);
        return;
    }
    if (remaining.length() == 0) return;
    mystery(prefix + remaining.charAt(0), remaining.substring(1), k-1);
    mystery(prefix, remaining.substring(1), k);
}
```

- a) What is the value returned by the following invocation? Trace your program. `mystery("", "CSEN", 3)`
- b) What does the above method do? Give an concise verbal description of how the value returned by `mystery("", s, k)` is related to the values of the parameters `s` and `k`.

Exercise 5-19 Search

Write a recursive method `search()` to search for a `char` inside a `String` and returns its position inside the `String`. The method should returns -1 if the `char` is not in the `String`.

Use the following `main` method to test your program:

```
public static void main(String [] args) {
    System.out.println(search("example", 'a'));
}
```

Hint:

- `"Hello".substring(1)` returns `"ello"`.
- `"Hello".substring(1,4)` returns `"ell"`.
- `"Hello".substring(0,s.length()-1)` returns `"Hell"`.

Exercise 5-20 Put At Front - Final Spring 2013
To be discussed in the Lab

Write a recursive method `putAtFront` that takes two parameters, a string `s` and a character `c`. The method returns a string with all occurrences of `c` placed at the front of the string and all other characters afterwards, in the same order they appear in the input string. If `c` does not exist, then the output string is the same as `s`. If `c` is at the beginning of `s`, and it does not appear in `s` any more time, then the output string is also the same as `s`.

The following list illustrates 4 different calls and the correct return value from calling the method each time.

```
Call: putAtFront("sce", 'c');
Return: cse
// note how c is at the front of the returned
// string and the remaining characters afterwards.
Call: putAtFront("static", 't');
Return: ttsaic
// here t appears twice in input string s.
// In the returned string, both are at front,
```

```
// and the remaining afterwards.
Call: putAtFront("banana", 'a');
Return: aaabnn
Call: putAtFront("java", 'j');
Return: java
Call: putAtFront("ALL", 'L');
Return: LLA
```

Exercise 5-21 PerfectRec

A positive integer is said to be perfect if the sum of its factors (excluding the integer itself) is that integer. For example, 6 is perfect, since the numbers that divide into it exactly are 1, 2, 3, and 6, and the sum of 1, 2, and 3 is itself 6.

Write a recursive method `perfectRec` to calculate the sum of divisors of `n` and use it to output whether `n` is perfect or not.

Exercise 5-22 Look And Say

The look-and-say sequence is the sequence of integers beginning as follows:

1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, ...

To generate a member of the sequence from the previous member, read off the digits of the previous member, counting the number of digits in groups of the same digit. For example:

- 1 is read off as “one one” or 11.
- 11 is read off as “two ones” or 21.
- 21 is read off as “one two, then one one” or 1211.
- 1211 is read off as “one one, then one two, then two ones or 111221.
- 111221 is read off as “three ones, then two twos, then one one” or 312211.

Write a recursive Java method `lookAndSay` and prints the first n^{th} terms of this sequence. **Hint:** you can create a helper method that takes a String as a parameter acting as the i^{th} term of the sequence and returns a String representing the $i^{th}+1$ term..

Exercise 5-23 Recursion Tracing 3

Given the following method:

```
public static void mystery( int n) {
    if (n/2 == 0 ) {
        System.out.print(n);
    } else if (n % 2 == 0) {
        System.out.print("(" + n + " + ");
        mystery(n - 1);
        System.out.print(")");
    } else {
        System.out.print("(");
        mystery(n - 1);
        System.out.print(" + " + n + ")");
    }
}
```

- a) What does the method display for the following call:

```
mystery(6);
```

Trace your method using a stack.

- b) What does the method displays for any integer?

Exercise 5-24 MergeRec

Write a recursive method mergeRec that given two strings displays the characters of the given strings in an alternating way. Note that the two strings could be of different length. Note that you are not allowed to use any additional strings.

Once you execute the following main method

```
public static void main(String[] args) {  
    String a = "hlo";  
    String b = "el";  
    mergeRec(a,b);  
}
```

the following should be displayed:
h e l l o

```
public static void mergeRec(String a, String b) {
```