King Fahd University of Petroleum & Minerals

Information & Computer Science Department

ICS-201 INTRODUCTION TO COMPUTER SCIENCE

**Lab #6: Recursion**

**Objectives:**

* To understand what is Recursion?
* Recursive Algorithms.
* Conversion between Recursive and Iterative algorithms.
* Problem Solving using Recursion.

**Recursion**

* Recursion is a powerful concept that helps to simplify the solution of complex problems. Recursion means defining something in terms of itself.
* A recursive method is a method that calls itself directly or indirectly.
* A recursive method has two major steps:
  + Recursive step in which the method calls itself
  + Base step which specifies a case with a known solution
* The method should select one of two steps based on a criteria
  + Executing recursive algorithms goes through two phases:
  + Expansion in which the recursive step is applied until hitting the base step.
  + “Substitution” in which the solution is constructed backwards starting with the base step.
* Recursion can be removed by replacing the selection structure with a loop.
* If some data need to be stored for processing after the end of the recursive step, a data structure is needed in addition to the loop.
* The data structure varies from a simple string or an array to a stack.

**Example 1**

An example of a recursive function is factorial

factorial(0) = 1 (Base Case)

factorial(n) = n \* factorial (n-1 ) (Recursive Step)



**Removing Recursion**

# Recursion can be removed by replacing the selection structure with a loop

# If some data need to be stored for processing after the end of the recursive step, a data structure is needed in addition to the loop.

**Example 2**

Write a recursive method to convert a String representing a decimal number to its binary equivalent.

* The algorithm (in pseudo-code) for converting a decimal integer into a binary integer as follows:
  1. If the integer is 0 or 1, its binary equivalent is 0 or 1.
  2. If the integer is greater than or equal to 2 do the following:
  3. Divide the integer by 2.
  4. Separate the result into a quotient and remainder.
  5. Divide the quotient again and repeat the process until the quotient is zero.
  6. Write down all remainders in reverse order as a string.
  7. This string is the binary equivalent of the given integer.



**Example 2-Complete program**

# 

**Exercise 4**

**Iterative Solution**

# 

# **Exercise 1**

Write a java method that computes ab recursively. Both a and b are positive integers.

*Hint:* ab = a \* ab-1

# **Exercise 2**

Write a recursive method to rotate a String by N characters to the left. For example, rotateLeft("ICS201", 2) should return "S201IC"

# **Exercise 3**

Solve the question in Lab 5 Example 4 using recursion instead of a loop.

A woman had just filled the cookie jar when the 3 children went to bed. That night one child woke up, ate half of the cookies and went back to bed. Later, the second child woke up, ate half of the remaining cookies, and went back to bed. Still later, the third child woke up, ate half of the remaining cookies, leaving 3 cookies in the jar. Develop an algorithm to find how many cookies were in the jar to begin with?