CSIT121 Lesson 8 Exercise

1. Write a recursive program that implements this definition of ***square numbers*** for a positive value of n > 1.
   square(n) = square(n-1) + 2n -1
   Aside: where did this crazy definition of square come from?
   Easy: this is just algebra:
   $(n-1)^2 = n^2 – 2n + 1$
   rearrange to get:
   $n^2 = (n-1)^2 + 2n - 1$

2. Write a recursive program that implements this definition of ***cube numbers*** (for positive integers):

   cube(n) = cube(n-1) + 3*(square(n)) - 3*n + 1

3. Multiplication of non-negative integers can be defined recursively in terms of addition:
   mult(n,0) = 0
   mult(n,m) = n+mult(n,m-1)
   Using recursion write a method mult which implements such a function.

4. Exponentiation of non-negative integers can be defined recursively in terms of Multiplication:
   $n^0 = 1$
   $n^m = n*(n^{m-1})$
   Using recursion write a method power which implements such a function.

5. The Syracuse Sequence starting with 14 goes like this:
   14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
   The rule is as follows: if n is even then the next number in the sequence is n/2 and if n is odd the next number is 3n + 1.
   (note: if n is even, n/2 will not have a remainder)
   The sequence stops at 1.
   Using recursion, write a program such that for all positive integers, n prints out the Syracuse sequence starting with n.