```
#|*****************************************************
Faisal Ibrahim                                  Homework 1
CISC 3140-ET6                                Date: 09/17/19
Prof. Murray Gross
*****************************************************|#
#| This is the Pyramid program which calculates number of
oranges on pyramid. The rule is that on each step n ther
are 2^n oranges on that level. So on ground level there
is 2^0 orange and on 1st step there are 2^1. So my program
calculates total number of oranges on the pyramid. |#

; PROGRAM CODE:
; I defined the function as "pyramid"
(define (pyramid steps) ; it takes one perameter
                        ; in my case it is called steps
                        ; so each step you take u get a different
  (if (= steps 0) ; at the bottom steps is zero so the oranges is 1
      0 ; I was making a mistake of writing this as "(1)" which
        ; is wrong since in scheme everything is a list and first
        ; one is always a function and last one is null or '().
#| In the following line of code we are doing 3 things
      ; 1. get the exponential value associated with that step
      ; 2. make a recursive call to the pyramid function with a
      ;    parameter value of one step less.
      ; 3. Add them both together                              |#
      (+ (expt 2 (- steps 1)) (pyramid (- steps 1)))))

"TESTS WITH DIFFERENT HEIGHT PYRAMIDS"
; pyramid of height 5 steps
"height 5 pyramid: "
(pyramid 5); answer should be 31
; pyramid of height 3 steps
"height 3 pyramid: "
(pyramid 3); answer should be 7
; pyramid of height 14 steps
"height 14 pyramid: "
(pyramid 14)
; pyramid of height 1 steps
"height 1 pyramid: "
(pyramid 1)
; hmmmmmmmmm, that's intersting why are we getting 3 oranges,
; shouldn't it be 1. Point to be noted at the step value zero
; we return 1 so our program is design structure is in a way
; that we get 1 orange at ground floor. If we change our
;  return value in the if statement's first argument to 0
; from 1, then it is a different case
#| NOTE: To resolve the above cited issue, I made some changes
   in the code:
              1. return value for null? is 0 insted of 1
              2. line 30 "exp 2 steps" is replaced with
                 "exp 2 (- 1 steps)
   This change gives me 1 orange at first step instead of 1
   on the ground. |#
; RESULT --> match with expected value
```