Rajat Arora (40078146) **Arash Manpreet Singh (40095187)**

Programming Questions:

b.)

The Mutiple tetranacci have a exponential time complexity because each call of tetranacci Number, creates multiple branches in the stack , that's create an exponential growth in the number of recursive calls.

The linear tetranacci calls the function recursively n times till it reach the base case . Making only one call each time , it removes the problem that was occurring in multiple recursion. Basically , linear recursion keeps a track of the calculated values so we don't have to calculate them again ,whereas in multiple recursion , same value is calculated multiple times.

c.)

The linear recursion is actually the tail recursive algorithm because the method has a recursive call as its last step.

Pseudo Code for multiple recursion.

Algorithm tetranacci(value)

Input : an integer value

Output : Tetranacci number at the given index.

If value < 3 **then**

    Return 0

If value <- 3 **then**

    Return 1

Else

    Return (tetranacci(value-1)+ tetranacci(value-2)+ tetranacci(value-3)+ tetranacci(value-4))

Pseudo Code for Linear recursion.

Algorithm tetranacciNumber(value, I,j,k,l)

Input: an integer value, and I,j,k,l (initial terms).

Output: A Tetranacci number whose index corresponds to the input number

if number ← 0 **then**

    return i

if number ← 1 **then**

    return j

if number ← 2 **then**

    return k

if number ← 3 **then**

    return l

return tetranacciNumber(value - 1, j, k, l, I+j+k+l)