**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**

**DEPARTAMENTO ACADÉMICO DE INGENIERÍA INFORMÁTICA**

**RECURSIVIDAD**

**EJERCICIOS RESUELTOS**

**(Mgt. Zonia Acurio Usca)**

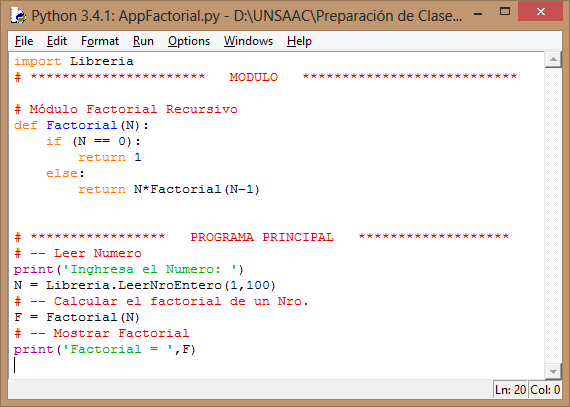
**Ejercicios de acuerdo a la clase de recursividad**

Se puede tener varias clasificaciones posibles, según el criterio elegido. Se consideran los siguientes criterios:

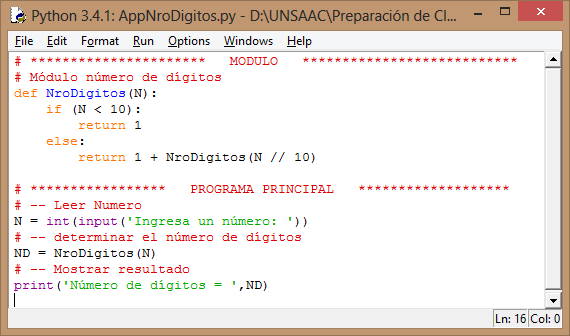
**Por el principio de inducción requerido**

***Recursividad estructural*.-** Se da cuando la reducción de la complejidad del problema se logra decrementando en uno el parámetro que controla la recursión. Un ejemplo es la función factorial.

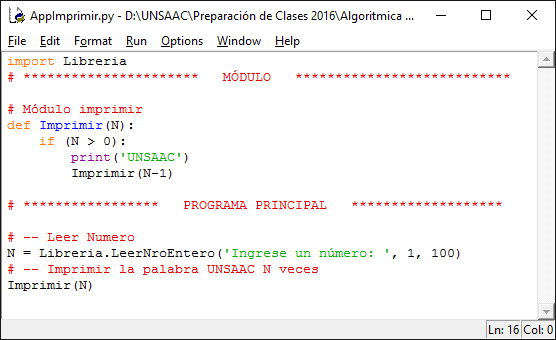
**1.- Escribir una aplicación para calcular el factorial de un número entero positivo**



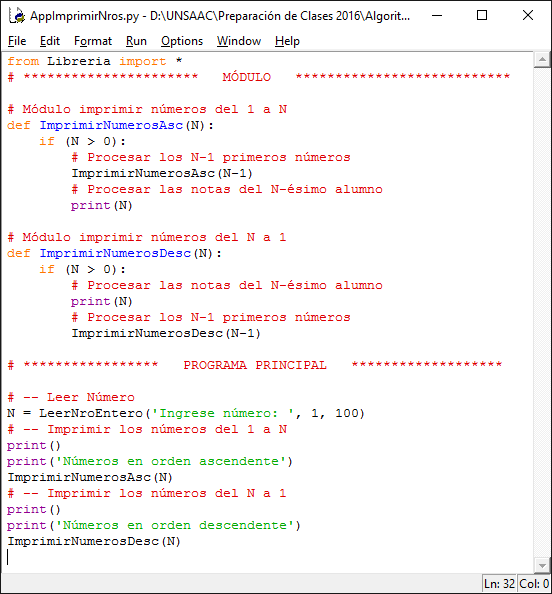
**2.- Escribir una aplicación para determinar el número de dígitos de un número entero positivo**



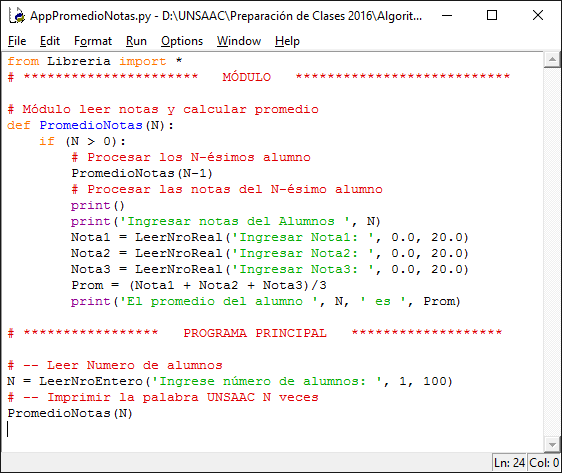
**3.- Escribir N veces la palabra UNSAAC**



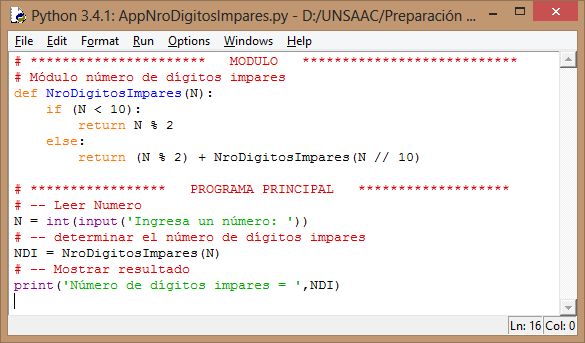
**4.- Escribir una aplicación que muestre los números del 1 al N, en orden ascendente y descendente.**



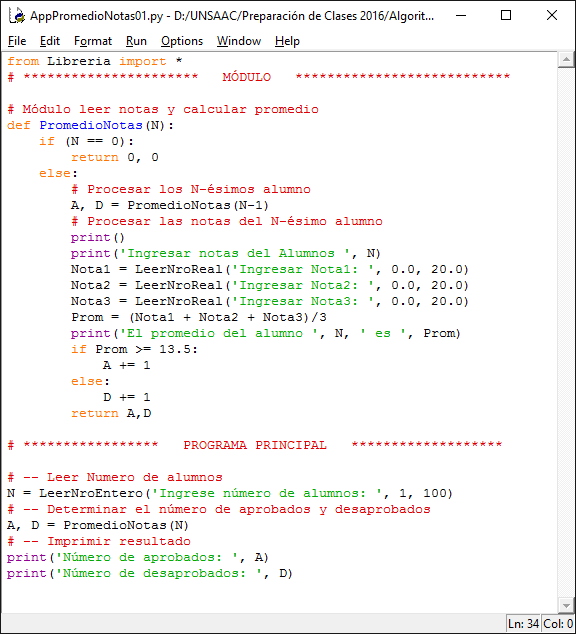
**5.- Se tiene N alumnos, para cada alumno se tiene tres notas. Escribir una aplicación para calcular el promedio de cada alumno.**



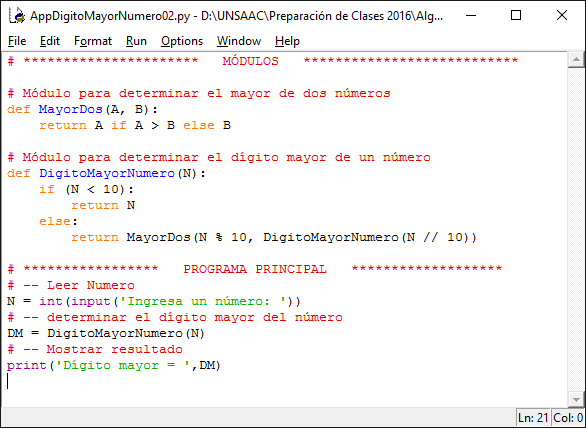
**6.- Escribir una aplicación para determinar el número de dígitos impares de un número entero positivo**



**7.- Se tiene N alumnos, para cada alumno se tiene tres notas. Escribir una aplicación para calcular el promedio de cada alumno y determinar el número de aprobados y desaprobados.**

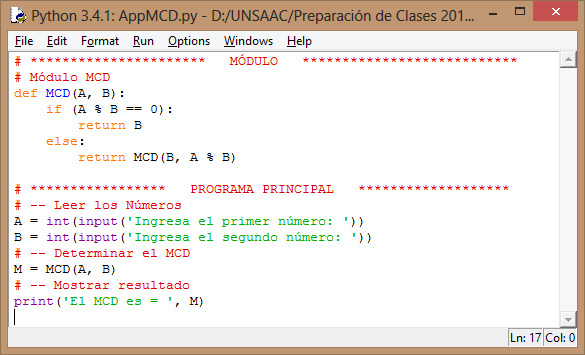


**8.- Escribir una aplicación para determinar el dígito mayor de un número entero mayor a cero**



***Recursividad no estructural*.-** Se da cuando la reducción de la complejidad del problema se logra con expresiones distintas a decrementos en uno, aplicadas al parámetro que controla la recursión. Un ejemplo es la función Máximo Común Divisor.

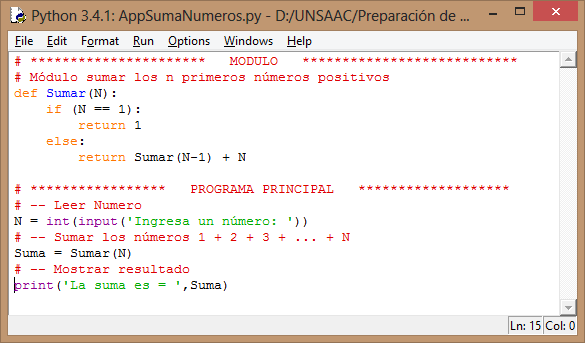
**9.- Escribir una aplicación recursiva para determinar el MCD de dos números**



**Por el nivel de complejidad**

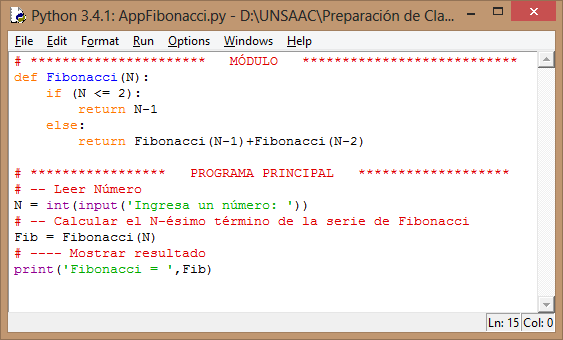
***Recursividad Lineal.-***El cuerpo del subprograma contiene una llamada recursiva. Este es el caso más simple de recursividad. Un ejemplo es la función factorial. Generalmente los problemas de Recursividad estructural son también de este tipo, tal como se vio en los ejemplos anteriores.

**10.- Escribir una aplicación recursiva para determinar la suma de los primeros N números enteros positivos**

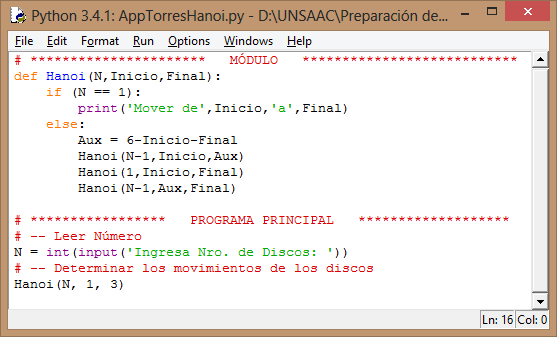


***Recursividad no lineal.***

**11.- Escribir un programa recursivo para determinar el n-ésimo término de la serie de Fibonacci.**



**12.- Escribir una aplicación para determinar los movimientos que se deben efectuar en el problema de las Torres de Hanoi.**



***Recursividad mutua.-*** Este es un caso de recursividad indirecta, y se da cuando dos o más subprogramas se llaman entre sí.

**13.- Escribir un subprograma recursivo para evaluar la siguiente expresión.**

**n-1**

C(n) = (2/ **n**) **Σ**  C(**i**) + **n** donde: C(0) = 1

**i** = **0**

**Modelo**

Caso base:

**1** Si **n** = 0

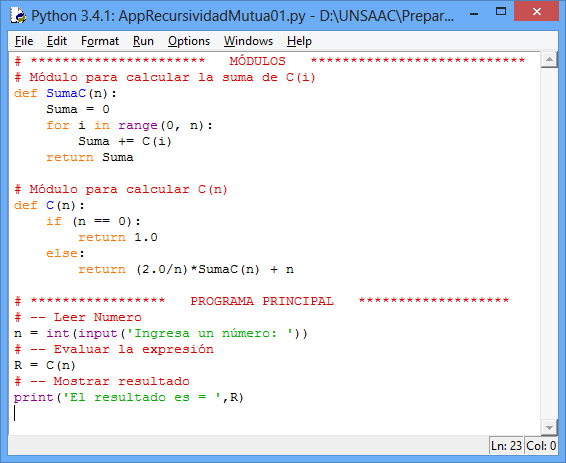
C(**n**) = Caso recurrente:

**n-1**

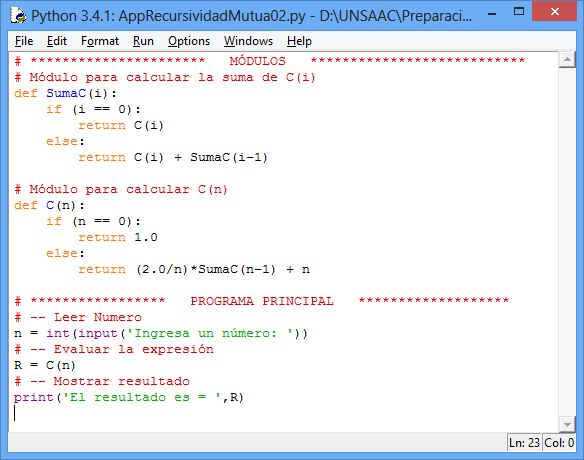
(2/ **n**) **Σ**  C(**i**) + **n** Si **n** > 0

**i** = **0**

**Primer algoritmo**



**Segundo algoritmo**



**14.- Escribir un subprograma recursivo para evaluar la siguiente expresión:**

**n -**1

T(**n**) = **∑** T( **i** ) T( **n-i** ) Donde: T(1)=T(2)=1.

**i** =1

**Modelo**

Caso base:

**1** Si **n** <= 2

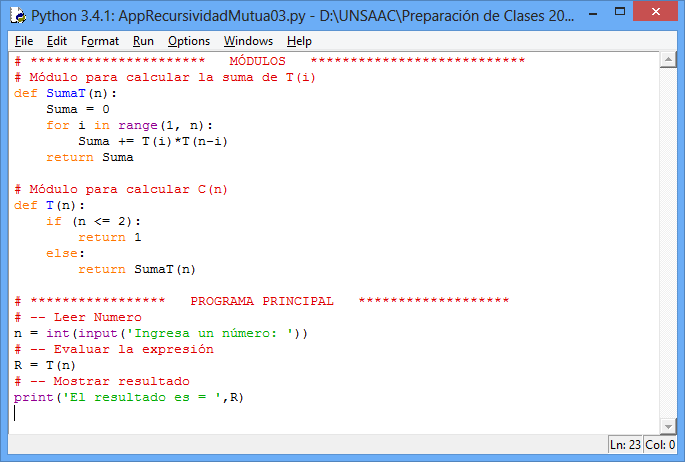
C(**n**) = Caso recurrente:

**n -**1

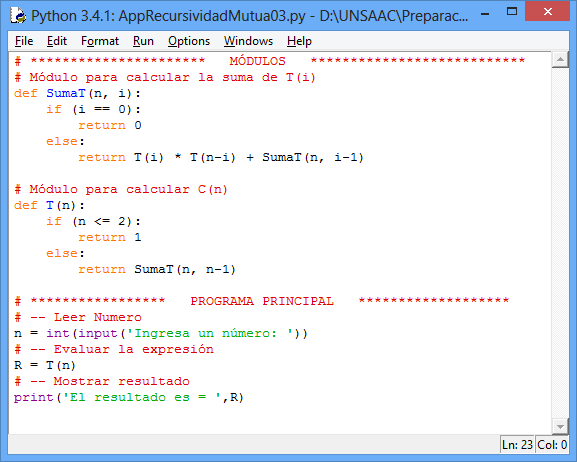
**∑** T( **i** ) T( **n-i** ) Si **n** > 2

**i** =1

Primer algoritmo



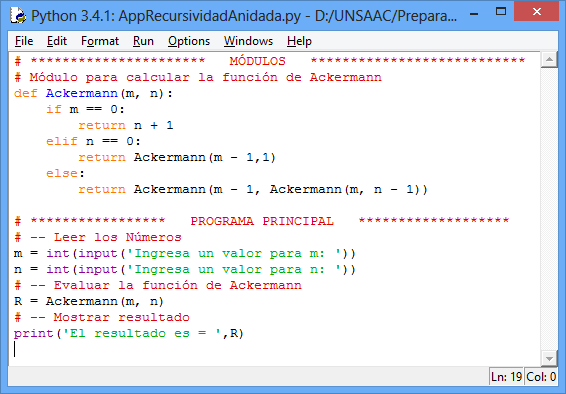
Segundo algoritmo



***Recursividad anidada.-*** Es cuando la función recursiva recibe como parámetro a otra función recursiva. La complejidad de este tipo de recursividad es más alta y generalmente resulta mucho más difícil de calcular.

**15.- Escribir un programa recursivo para la función de Ackermann.**


  A(m,n)=
    \begin{cases}
     n+1,               &\mbox{si }m=0;
    \\
     A(m-1, 1),         &\mbox{si }m>0\mbox{ y }n=0;
    \\
     A(m-1, A(m, n-1)), &\mbox{si }m>0\mbox{ y }n>0
    \end{cases}
  



**PROBLEMAS**

**16.- Escribir un subprograma recursivo para convertir un número entero positivo de base 10 a base 2.**

Modelo

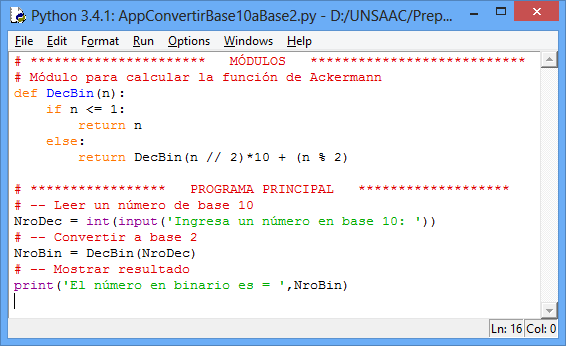
Caso base:

n Si **n** <= 1

DecBin(**n**) =

Caso recurrente:

DecBin(**n** DIV 2)\*10 + (**n** MOD 2) Si **n** > 1



**17.- Escribir un subprograma recursivo para calcular coeficientes binomiales.**

Modelo

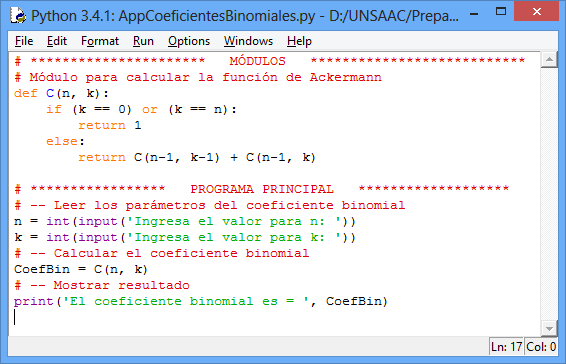
Caso base:

1 Si **k** = 0 o **k = n**

C(**n,k**) =

Caso recurrente:

C (**n**-1,**k**-1) + C(**n**-1,**k**) Si **n** > **k** > 0



**18.- Un robot puede desplazarse en línea recta dando pasos de uno o dos metros. Escribir un subprograma recursivo que determine de cuantas maneras distintas puede desplazarse el robot sobre “n” metros.**

Modelo

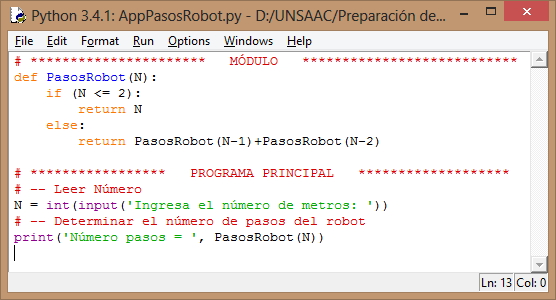
Caso base:

**n** Si **n** <= 2

PasosRobot(**n**) =

Caso recurrente:

PasosRobot (**n**-1)+ PasosRobot (**n**-2) Si **n** > 2



**19.- En una base Militar en el desierto hay un Jeep y n bidones de gasolina. El tanque de gasolina del Jeep tiene la capacidad para un bidón y además puede cargar otro bidón. El Jeep puede recorrer D Km con un bidón de gasolina. Escribir un subprograma que calcule la distancia máxima que puede recorrer el Jeep.**

Modelo

Caso base:

**D** Si **n** = 1

Distancia(**D,n**) =

Caso recurrente:

**D/** (**n**-1)\* 2-1+Distancia(**D,n**-1) Si **n** > 1

