EJERCICIO 1: Sea $n$ un número natural y considere el siguiente código en Python que define la función recursiva $F(n)$:

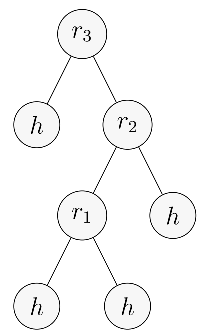
```
Def F(n):
    if n == 0:
        return 0
    else:
        return n^2 + F(n−1)
```

a) Escriba el paso a paso de $F(3)$.

b) Demuestre que
$F(n) = \frac{n(n+1)(2n+1)}{6}$.

EJERCICIO 2: Sea $A$ un árbol binario y considere el siguiente código en Python que define la función recursiva $F(A)$:

```
Def F(A):
    if A.right == None:
        return [0]
    elif A.label == '¬':
        return [x + 1 for x in F(A.right)]
    elif A.label in ['∧', '∨', '→', '↔']:
        return [x + 1 for x in F(A.left)]
                  + [x + 1 for x in F(A.right)]
```




a) Escriba el paso a paso de $F[r_3]$ para el árbol $r_3$ definido en la figura de arriba.

b) Demuestre por inducción estructural sobre $A$ que $F[A]$ es una lista que contiene el número de aristas de cada rama del árbol de izquierda a derecha.

EJERCICIO 3: Sea $A$ una fórmula representada como un árbol y asuma la definición de las siguientes funciones:

- $C\_Bin(A)$: Número de ocurrencias de conectivos binarios en $A$.
- $Negs(A)$: Número de ocurrencias de negaciones en $A$.
- $Atom(A)$: Número de ocurrencias de letras proposicionales en $A$.
- $Inorder(A)$: Cadena que representa la notación inorder de $A$.
- $Len(c)$: Cantidad de símbolos en la cadena $c$.

Demuestre que:

$$Len(Inorder(A)) = 4 * Atom(A) + Negs(A) - 3$$

[Ayuda: Demuestre primero por inducción estructural que

$$Len(Inorder(A)) = 3 * C\_Bin(A) + Negs(A) + Atom(A)$$

y que $C\_Bin(A) = Atom(A) - 1$.]

Ejercicio 4: Considere la siguiente definición alternativa de la valuación de una fórmula:

Def $Vmod2_I(f)$:
  Si $f$. right == null:
    Retornar $I(f.\text{label})$
  Si no, si $f.\text{label} == \neg$:
    Retornar $Vmod2_I(f.\text{right}) + 1 \pmod 2$
  Si no, si $f.\text{label} == \wedge$:
    Retornar $Vmod2_I(f.\text{left}) \times Vmod2_I(f.\text{right})$
  Si no, si $f.\text{label} == \vee$:
    Retornar $Vmod2_I(f.\text{left}) \times Vmod2_I(f.\text{right}) + Vmod2_I(f.\text{left}) + Vmod2_I(f.\text{right}) \pmod 2$
  Si no, si $f.\text{label} == \rightarrow$:
    Retornar $Vmod2_I(f.\text{left}) \times Vmod2_I(f.\text{right}) + Vmod2_I(f.\text{left}) + 1 \pmod 2$

Demuestre que $V_I(A) = Vmod2_I(A)$ para toda fórmula $A$.