Ejercicio 1 El problema se resuelve recursivamente de la siguiente manera: Se define una función en la que recibe como entrada la cadena y retorna si es palíndromo o no. Para cada cadena ingresada, se pregunta si el último caracter es igual al primero. En caso de que no sean iguales entonces retorna que no es palíndromo. En caso de que sí sean iguales, retorna la función evaluada en la cadena obtenida al quitarle la primer y último caracter. Finalmente, los casos iniciales son cuando la cadena tenga un caracter o ninguno, en ambos casos la cadena sí es un palíndromo.

Ejercicio 2 Sabemos que $F_n = F_{n-1} + F_{n-2}$ donde F_n es el enésimo número de Fibonacci y que $F_0 = 0$ y que $F_1 = 1$. Entonces definimos una función que pida de entrada un entero n y que retorne el valor de la función evaluada en $n-1$ mas el valor de la función evaluada en $n-2$ si $n \geq 2$. Para los casos iniciales, en caso de que n sea 0 entonces retorne 0, y si n es igual a 1 entonces retornará 1.

Ejercicio 3 Sabemos que $\binom{N}{K} = \binom{N-1}{k} + \binom{N-1}{K-1}$. Usaremos una función en la cual se ingresen dos valores N y K y retorne la suma de la función evaluada en $N-1$ y $K-1$ mas la función evaluada en $N-1$ y K. Los casos iniciales van a ser si N o K son 0 y en este caso la función debe de retornar el valor de 1.

Ejercicio 4 La forma de resolver el ejercicio es crear una función en la cual se pida de entrada la cadena. La función sumará cada uno de los números dentro de la cadena, y cuando encuentre un paréntesis, buscará su respectivo paréntesis de salida y sumar esta vez la función aplicada a la subcadena.

Ejercicio 5 Sea H_n la respuesta para n discos. Sabemos que recursivamente $H_n = 2H_{n-1} + 1$. De aquí se puede demostrar que $H_n = 2^n - 1$.