

_4bucles/_1torres_hanoi.py

```

1  """Rekursividad:
2  Tienes a tu disposición un conjunto de discos numerados del 1 al N y tres torres
3  etiquetadas como A, B y C. La torre A contiene inicialmente todos los discos
4  apilados en orden descendente, desde el disco N en la parte inferior hasta el
5  disco 1 en la parte superior.
6  Tu tarea es implementar una solución recursiva para mover todos los discos
7  desde la torre A hasta la torre C, siguiendo las reglas clásicas de las Torres de
8  Hanoi:
9  1. Puedes mover un disco a una torre adyacente.
10 2. Solo puedes mover un disco a la cima de otra pila si esa pila está vacía o si
11 el disco superior es más grande que el disco que estás colocando.
12 Debes definir una función llamada torres_de_hanoi(n, origen, destino, auxiliar)
13 que, dado el número total de discos n y las torres de origen, destino y auxiliar,
14 imprima los pasos necesarios para lograr el movimiento de todos los discos
15 desde la torre A hasta la torre C.
16 A continuación un ejemplo de una posible entrada y salida de la solución:
17 Entrada Salida
18 - N de discos
19 - N de torres
20 - Torres : origen, destino, auxiliar
21 Mover disco de la torre A a la torre D
22 Mover disco de la torre A a la torre B
23 ...
24 """
25 # Ejercicio1 :
26 # Tienes a tu disposición un conjunto de discos numerados del 1 al N y tres torres
27 etiquetadas como A, B y C. La torre A contiene inicialmente todos los discos apilados en
28 orden descendente, desde el disco N en la parte inferior hasta el disco 1 en la parte
29 superior.
30 # Tu tarea es implementar una solución ***recursiva*** para mover todos los discos desde la
31 torre A hasta la torre C, siguiendo las reglas clásicas de las Torres de Hanoi:
32
33 # 1. Puedes mover un disco a una torre adyacente.
34 # 2. Solo puedes mover un disco a la cima de otra pila si esa pila está vacía o si el disco
35 superior es más grande que el disco que estás colocando.
36
37 # Debes definir una función llamada **`torres_de_hanoi(n, origen, destino, auxiliar)`** que,
38 dado el número total de discos **`n`** y las torres de origen, destino y auxiliar, imprima
39 los pasos necesarios para lograr el movimiento de todos los discos desde la torre A hasta la
40 torre C.
41
42 def mover_disco(desde,hacia,disco):
43     print(f'Mover disco {disco} de la torre {desde} hacia la torre {hacia}')
44
45 def torres_de_hanoi(n,origen,destino,auxiliar):
46     #caso base /// recursividad
47     if n == 1 :
48         mover_disco(origen,destino,n)
49         return
50     #continuando con la recursividad
51     torres_de_hanoi(n-1,origen,auxiliar,destino)

```

```
45 mover_disco(origen,destino,n)
46 torres_de_hanoi(n-1,auxiliar,destino,origen)
47
48
49 torres_de_hanoi(3,'Origen','Destino','Auxiliar')
50
51 def torres_de_hanoi(n, origen, destino, auxiliar):
52     # Caso base: si solo hay un disco, moverlo directamente al destino
53     if n == 1:
54         print(f"Mover disco de la torre {origen} a la torre {destino}")
55         return
56
57     # Mover los n-1 discos desde la torre de origen a la torre auxiliar
58     torres_de_hanoi(n - 1, origen, auxiliar, destino)
59
60     # Mover el disco restante desde la torre de origen a la torre de destino
61     print(f"Mover disco de la torre {origen} a la torre {destino}")
62
63     # Mover los n-1 discos desde la torre auxiliar a la torre de destino
64     torres_de_hanoi(n - 1, auxiliar, destino, origen)
65
66 # Ejemplo de uso
67 n = 3 # Número de discos
68 origen = 'A' # Torre de origen
69 destino = 'C' # Torre de destino
70 auxiliar = 'B' # Torre auxiliar
71
72 # Llamar a la función torres_de_hanoi
73 torres_de_hanoi(n, origen, destino, auxiliar)
74
```