

Pregunta 1. En el "juego de la oca" un jugador avanza su ficha por un tablero lineal de 63 casillas, lanzando en cada turno dos dados de 6 caras simultáneamente. El objetivo del juego es llegar a la casilla 63, pero sin pasarse. Esto es, si el jugador no cae exactamente en la meta, entonces se devuelve tantas casillas como las que se pasó. Además, en el tablero existen casillas especiales que se llaman "ocas" (que corresponden a la 9, 18, 27, 36, 45, 54 y 63) y hacen que el jugador se mueva automáticamente desde una "oca" a la siguiente.

A)Escriba la función de nombre **mover** (sin la receta de diseño) que recibe como entrada el número de la casilla en que está el jugador y devuelve el número de la casilla en la que queda al terminar su turno.

Ejemplos:

- Si estaba en la casilla 1 y saca 9, entonces se mueve a la 10
- Si estaba en la 1 y saca 8, entonces se mueve a la 9, pero como es oca, termina en la 18 (que es la oca que sigue al 9)
- Si estaba en la 55 y saca 10, entonces se mueve a la 61, porque al llegar a la 63 le sobraron 2 casillas por mover, y entonces se devuelve esas 2 casillas
- Si estaba en la 62 y saca 10, entonces cae en la 54, pero como es oca, termina en la 63

Indicación. Para simular el lanzamiento de un dado use la función `randint` del módulo `random`. Por ejemplo, `random.randint(13,21)` devuelve un número entero N al azar, tal que $13 \leq N \leq 21$.

B)Escriba la función recursiva de nombre **cantidadDeTurnos** (sin la receta de diseño) que devuelve la cantidad de turnos (por ejemplo 13) que tuvo que hacer un jugador desde el inicio del juego hasta llegar a la meta, simulando los movimientos en cada turno con la función `mover`.

Nota. Si lo estima necesario, puede agregar parámetro(s) a la función `cantidadDeTurnos`.

Pregunta2. Un conjunto de dígitos entre 1 y 9 se puede representar por un número entero. Por ejemplo el conjunto {3,1} se representa por el entero 31 (o también por el entero 13) y el conjunto vacío {} se representa por un 0. Al respecto, el módulo conjunto contiene las funciones indicadas en los siguientes ejemplos:

Ejemplo1	resultado	Ejemplo2	resultado
esConjunto(312)	True	esConjunto(313)	False (tiene repetidos)
pertenece(1,312)	True (dígito 1 está en 312)	pertenece(4,312)	False (dígito 4 no está)
interseccion(312,4352)	32	Intersección(312,456)	0 (vacío)
union(312,4352)	31245	union(312,456)	312456
resta(312,4352)	1	resta(4352,312)	45
cardinal(4352)	4	cardinal(0)	0

A)(2 pts) Escriba un programa para leer dos enteros que representan dos conjuntos y escriba el número entero que representa el conjunto con la diferencia simétrica entre los dos conjuntos y su cardinal (cantidad de elementos).

Notas.

- El programa debe usar las funciones del módulo conjunto (sin programarlas).
- La diferencia simétrica entre dos conjuntos se calcula como la resta entre la unión y la intersección de los conjuntos.
- En caso que alguno de los números no represente un conjunto válido se debe escribir el mensaje 'conjunto erroneo' y no calcular la diferencia simétrica.

B)(4 pts) Escribir la **función recursiva interseccion** del módulo conjunto con la receta de diseño completa (incluyendo un par de pruebas).

Nota. Solo puede usar la función de nombre **pertenece** del módulo conjunto, sin necesidad de programarla.