21.Definir una función para obtener la matriz triangular superior (incluyendo la diagonal principal) de una matriz cuadrada que está representada como una lista de listas.

Para resolver este ejercicio hice un for para poder recorrer las posiciones de la lista de listas. Como es cuadrada, obtengo la cantidad de elementos en L usando count de L y genero el rango del for con ese número.

Luego, en cada posición del for, dada por la variable x, tomo el elemento en la x-ésima posición y tomo los últimos elementos según el cálculo (rango L – x). Por ejemplo, cuando x sea 0, voy a tomar todos los elementos de la lista en la posición 0. Cuando sea 1 voy a tomar todos los elementos menos el primero de la lista en la posición 1 y así sucesivamente, como se puede ver en el ejemplo.

(defn triang-sup [L] (for [x (range (count L))] (take-last (- (count L) x) (nth L x))))

(triang-sup '((1 2 3 4 5) (6 7 8 9 10) (11 12 13 14 15) (16 17 18 19 20) (21 22 23 24 25)))

22.Definir una función para obtener la diagonal principal de una matriz cuadrada que está representada como una lista de listas.

Este ejercicio lo resolví de manera similar al anterior. Genero el for de la misma manera que en el anterior, pero en éste únicamente necesito el elemento en la diagonal principal, por lo que, por ejemplo, del elemento en la posición 0 de la lista de listas necesito solo el elemento en la posición 0.

Entonces, tomo el x-ésimo elemento de la x-ésima lista y resulta en lo pedido, como se ve en la parte final de la función: (nth (nth L x) x).

(defn diag-princ [L] (for [x (range (count L))] (nth (nth L x) x)))

(diag-princ '((1 2 3 4 5) (6 7 8 9 10) (11 12 13 14 15) (16 17 18 19 20) (21 22 23 24 25)))