

Deberá entregar cada ejercicio en un .py diferente, con el nombre del archivo nombre-ej#.py (# será reemplazado por el número de ejercicio).

### Ejercicio 1:

Decimos que una matriz de  $n \times n$  es cuadrada si la cantidad de filas es igual a las columnas. La diagonal de una matriz cuadrada son todos los elementos que están ubicados en la posición  $i,i$  (donde  $i=1..n$ ,  $n$  la dimensión de la matriz). Una matriz se dice diagonalmente dominante si la suma de todos los elementos de la fila (excepto el de la diagonal) es menor que el elemento de la diagonal. Supongamos que las matrices se representan mediante una lista de listas (donde cada una de estas es una fila de la matriz).

Defina una función llamada `diagonal_dominante` que reciba como parámetro la matriz (una lista de listas) y retorne True si es diagonal dominante y False si no (se asume que la matriz está bien definida).

### Ejercicio 2:

Sea una matriz definida como el ejercicio anterior. Haga una función llamada `diagonal(matriz)` Que reciba una matriz y devuelva un string con los valores de la diagonal concatenados, pero en sentido inverso. Ej:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

deberá devolver el string "16 11 6 1"

### Ejercicio 3:

La siguiente es una lista de diccionarios. Cada uno de estos diccionarios representa la información de un alumno: identificador, nombre y las calificaciones que obtuvo en sus materias durante el semestre.

estudiantes = [

```
{ "id": 5, "nombre": "Alicia", "calificaciones": { "P1": 85, "TI3": 90 } },
{ "id": 2, "nombre": "Roberto", "calificaciones": { "P1": 78, "TI3": 88, "CORE": 95 } },
{ "id": 3, "nombre": "Carlos", "calificaciones": { "CORE": 92, "CALCULO": 80 } },
{ "id": 6, "nombre": "Ana", "calificaciones": { "CALCULO": 89, "ALGEBRA": 94, "CORE": 90, "MICRO": 87 } }
```

]

Realiza un programa que pida por pantalla el nombre de un alumno y muestre sus calificaciones en cada materia.

### Ejercicio 4

Escribir una función en Python que recibe como parámetro dos cadenas y retorna True si la primera cadena empieza con los caracteres de la segunda cadena (en el mismo orden) y False en caso contrario. Cada cadena recibida por parámetro puede contener mayúsculas o minúsculas. No se debe utilizar la función `startswith` de Python.

Ejemplo:

`comienza_por("radio","Ra")` debe retornar: True

`comienza_por("radio","ad")` debe retornar: False