

### Mejoramiento I Término 2003-2004. Septiembre 16, 2003

**Tema 1** (25 puntos) Un cuadrado mágico es una matriz cuadrada de orden  $n$ , tal que sus elementos son los números  $1, 2, 3, 4, \dots, n^2$ ; sin repeticiones y ubicados de tal manera que la suma de los elementos de cada fila es igual a la suma de los elementos de cualquier columna e igual a la suma de los elementos de las diagonales.

Por ejemplo, la matriz siguiente es un cuadrado mágico de tercer orden.

Escriba un programa que permita leer los elementos de una matriz  $n \times n$ , tal que  $n < 10$  y verifique si el cuadrado es mágico.

Rúbrica: ingreso y validación (5 puntos), suma de filas y columnas (5 puntos), suma diagonales (5 puntos), validar resultados (5 puntos). Solución integral (5 puntos)

Ejemplo:

4	9	2
3	5	7
8	1	6

Es un cuadrado mágico: 1

#### Propuesta de Solución:

El problema se separa en sus partes principales. Primero se revisa el número de veces que aparece cada número en el cuadrado. Luego se suma cada fila y columna, guardando los resultados en vectores. La suma de las diagonales se las hace aparte pues solo se necesita mover un índice; la diagonal principal tiene el mismo índice en la fila y columna; y la diagonal secundaria, en cambio la columna es decreciente.

Se validan los resultados parciales con las condiciones para que sea mágico y se muestra el resultado

Descripción	Python
Inicio	<code># ICM00794-Fundamentos de Computación - FCNM-ESPOL</code>
Se usará librerías numpy para matrices	<code># Mejoramiento I Término 2003</code>
Tamaño del cuadrado	<code># Tema 2. cuadrado magico</code>
Validando el tamaño menos de 10	<code># Propuesta: edelros@espol.edu.ec</code>
Inicializa cuadrado	<code>import numpy</code>
Ingresa los datos del cuadrado	<code>n=int(input("tamaño cuadrado: "))</code>
Verifica números repetidos	<code>while (n&gt;10):</code>
Tamaño de vector de contadores	<code>n=int(input("tamaño cuadrado(n&lt;10): "))</code>
Inicializa contadores en cero	<code>cuadrado=numpy.zeros(shape=(n,n),dtype=int)</code>
Bandera indica si hay más de uno	<code>for i in range(0,n,1):</code>
Revisa todo el cuadrado	<code>for j in range(0,n,1):</code>
Toma el valor de una casilla	<code>cuadrado[i,j]=input("cuadrado ["+str(i+1)+",""+str(j+1)+"]": )</code>
Cuenta el número de veces que aparece	<code># verifica números repetidos</code>
Si es más de uno, se cambia bandera	<code>m=n*n</code>
Se suman filas y columnas	<code>repetido=numpy.zeros(m+1,dtype=int)</code>
Inicializa los vectores suma	<code>masdeuno=0</code>
Se acumulan filas y columnas	<code>for i in range(0,n,1):</code>
Revisa las diagonales	<code>for j in range(0,n,1):</code>
Inicializa acumuladores	<code>k=cuadrado[i,j]</code>
Se suman las celdas de las diagonales	<code>repetido[k]=repetido[k]+1</code>
Se verifica con los resultados de los pasos anteriores	<code>if (repetido[k]&gt;1):</code>
Se supondrá que es mágico	<code>masdeuno=k</code>
Comprobando luego los resultados	<code>#suma de filas y columnas</code>
	<code>sfila=numpy.zeros(n,dtype=int)</code>
	<code>scolumna=numpy.zeros(n,dtype=int)</code>
	<code>for i in range(0,n,1):</code>
	<code>for j in range(0,n,1):</code>
	<code>sfila[i]=sfila[i]+cuadrado[i,j]</code>
	<code>scolumna[i]=scolumna[i]+cuadrado[j,i]</code>
	<code># Suma diagonales</code>
	<code>sdiagonal=0</code>
	<code>sdiagonal2=0</code>
	<code>for i in range(0,n,1):</code>
	<code>sdiagonal=sdiagonal+cuadrado[i,i]</code>
	<code>sdiagonal2=sdiagonal2+cuadrado[i,n-1-i]</code>
	<code># verifica condiciones magico</code>
	<code>magico=1</code>
	<code>if (masdeuno&gt;0):</code>
	<code>magico=0</code>
	<code>if not(sdiagonal==sdiagonal2):</code>
	<code>magico=0</code>

Se muestra el resultado	<pre>for i in range(0,n,1):     if not(sdiagonal==sfila[i]):         magico=0     if not(sdiagonal==scolumna[i]):         magico=0  #Salida print("El resultado es:") print(magico)</pre>
-------------------------	---

### Ejecución del algoritmo: cuadrdomagico.py

<pre>&gt;&gt;&gt; tamaño cuadrado: 3 cuadrado [1,1]: 4 cuadrado [1,2]: 9 cuadrado [1,3]: 2 cuadrado [2,1]: 3 cuadrado [2,2]: 5 cuadrado [2,3]: 7 cuadrado [3,1]: 8 cuadrado [3,2]: 1 cuadrado [3,3]: 6 El resultado es: 1</pre>	<pre>&gt;&gt;&gt; tamaño cuadrado: 3 cuadrado [1,1]: 2 cuadrado [1,2]: 9 cuadrado [1,3]: 4 cuadrado [2,1]: 3 cuadrado [2,2]: 5 cuadrado [2,3]: 7 cuadrado [3,1]: 8 cuadrado [3,2]: 1 cuadrado [3,3]: 6 El resultado es: 0</pre>
---	---