**1**. Este programa estudiará el nivel de coincidencia de dos matrices de números enteros. Dos matrices **A** y **B** (**N**x**N, N** constante conocida) tienen nivel **T** de coincidencia si hay exactamente **T** posiciones de **A** y **B** cuyos valores coinciden.

* Escribe un procedimiento **LeeMatriz** que lea una matriz **M**(**N**x**N**). *(0.1 puntos)*
* El procedimiento **LeeVector** leerá un vector de enteros **V(M)**, **M** constante conocida.*(0.1 puntos)*
* Diseña una función **CoincidenMatrices** que tome dos matrices **A** y **B** y nos diga cuántas posiciones de ambas matrices tienen valores coincidentes. *(1 punto)*
* Diseña ahora una función booleana **CoincideT** que tome dos matrices **A** y **B**, y un valor entero **T** y utilizando la función anterior nos diga si hay exactamente **T** posiciones coincidentes en las dos matrices**.** *(0,5 puntos)*
* La función **PosCoincidentes** tomará dos matrices **A** y **B** y un vector **V** y nos devolverá la posición del vector donde esté el número de coincidencias entre ambas matrices. Para hacerlo, deberás recorrer el vector y utilizar la función **CoincideT**. Si el número de coincidencias no está en el vector, devolverá 0 y si está repetido en el vector, la posición de la primera aparición. *(1 puntos)*

Completa el ejercicio con un programa principal que lea las dos matrices y el vector y nos diga si en alguna de las posiciones del vector está el número de coincidencias de las dos matrices. *(0,8 puntos)*

**def** LeeMatriz():  
 print(**'Escribe una matriz cuadrada de tamaño {0}x{0} por filas '**.format(n))  
 m1=[]  
 **for** i **in** range(n):  
 fila=input().split()  
 **for** j **in** range (n):  
 fila[j]=int(fila[j])  
 m1.append(fila)  
 **return** m1

**def** LeeVector():  
 v=[]  
 v=input(**'Escribe {} números'**.format(m)).split()  
 **for** i **in** range(m):  
 v[i]=int(v[i])  
 **return** v

**def** CoincidenMatrices(a,b):  
 num=0  
 **for** i **in** range(n):  
 **for** j **in** range(n):  
 **if** a[i][j]==b[i][j]:  
 num+=1  
 **return** num

**def** CoincideT(a,b,t):  
 **return** CoincidenMatrices(a,b)==t

**def** PosCoincidentes(a,b,v):  
 pos=0  
 i=0  
 **while** i<m **and** CoincideT(a,b,v[i])==**False**:  
 i+=1  
 **if** CoincideT(a,b,v[i]):  
 pos=i  
 **return** pos

n=4  
m=6  
mat1=LeeMatriz()  
mat2=LeeMatriz()  
vec=LeeVector()  
**if** PosCoincidentes(mat1,mat2,vec)==0:  
 print(**'Ninguna de las posiciones del vector coincide con el número de coincidencias'**)  
**else**: print(**'En la posición'** , PosCoincidentes(mat1,mat2,vec), **' está el número de coincidencias**

**2.** Escribe los siguientes procedimientos y funciones:

* Un procedimiento **LeeMatriz** que lea del usuario **N**x**N** números enteros (**N** constante conocida) y los almacene en una matriz. *(0.1 puntos)*
* Una función **ExtraeDiagonalPrincipal** que devuelva la diagonal principal de una matriz de entrada **M** en forma de vector. *(0.6 puntos)*
* Un procedimiento **ExtraeDiagonalSecundaria** que devuelva la diagonal secundaria de una matriz de entrada **M** en forma de vector. *(0.6 puntos)*
* Un procedimiento **ComparaVectores** que reciba dos vectores de entrada **V1** y **V2** de dimensión **N** y devuelva dos nuevos vectores (**PAR**, **IMPAR**), así como el número de elementos almacenados en cada uno de estos vectores (**TamPar**, **TamImpar**). Para ello, deben compararse los elementos de **V1** y **V2** elemento a elemento y guardar en los vectores la división entera del elemento en **V1** entre el de **V2**. La división se guardará en el vector **PAR** si el resto de la división entre los dos elementos es par y en **IMPAR** en otro caso. *(1,5 puntos)*
* Un procedimiento **EscribeVector** que reciba un vector y el número de elementos almacenados en él y muestre por pantalla dichos números en una única fila. *(0.2 puntos)*

Utilizando los procedimientos y funciones anteriores, crea un programa que lea dos matrices de números enteros, extraiga la diagonal principal de la primera matriz y la secundaria de la segunda y, posteriormente, compare las dos diagonales obteniendo los vectores **PAR** e **IMPAR**. Finalmente muestra el contenido de dichos vectores por pantalla. *(0.5 puntos)*

**def** LeeMatriz():  
 print(**'Escribe una matriz cuadrada de tamaño {0}x{0} por filas '**.format(n))  
 m=[]  
 **for** i **in** range(n):  
 fila=input().split()  
 **for** j **in** range (n):  
 fila[j]=int(fila[j])  
 m.append(fila)  
 **return** m  
  
**def** ExtraeDiagonalPrincipal(m):  
 dp=[0]\*n  
 **for** i **in** range(n):  
 dp[i]=m[i][i]  
 **return** dp

**def** ExtraeDiagonalSecundaria(m):  
 ds=[0]\*n  
 **for** i **in** range(n):  
 ds[i]=m[i][n-i-1]  
 **return** ds

**def** ComparaVectores(v1,v2):  
 par=[]  
 impar=[]  
 **for** i **in** range (n):  
 **if** (v1[i]%v2[i])%2==0:  
 par.append(v1[i]//v2[i])  
 **else**:  
 impar.append(v1[i]//v2[i])  
  
 **return** par,impar  
**def** EscribeVector(v,nv):  
 **for** i **in** range(nv):  
 print(v[i],end=**' '**)  
 print  
  
n=4  
mat1=LeeMatriz()  
mat2=LeeMatriz()  
dpmat1=ExtraeDiagonalPrincipal(mat1)  
dsmat2=ExtraeDiagonalSecundaria(mat2)  
vpar,vimpar=ComparaVectores(dpmat1,dsmat2)  
print(**'El vector par es '**)  
EscribeVector(vpar,len(vpar))  
print(**'El vector impar es '**)  
EscribeVector(vimpar,len(vimpar))