

Normas para la realización del examen:**Duración: 1.5 horas**

- Debe disponer de un documento oficial que acredite su identidad a disposición del profesor.
- Durante el tiempo que dura el examen, no debe conectarse a su plataforma de docencia desde ningún otro ordenador.
- Dispone de 30 minutos para decidir si no se presenta al examen (y no le cuenta convocatoria)

Se quiere representar las conexiones por avión entre un conjunto de n ciudades a través de una matriz cuadrada de tipo bool, atendiendo a lo siguiente:

- Cada ciudad viene identificada por un índice entero (0, 1, 2, etc)
- Si la ciudad i está conectada con la ciudad j , en la posición i, j de la matriz de bool se almacenará true (y false en otro caso). La matriz no tiene por qué ser simétrica ya que una ciudad i podría tener una conexión de ida con la ciudad j pero no tenerla de vuelta.
- Una misma ciudad nunca está conectada con ella misma.

Construya la clase Conexiones para representar esta información:

1. Defina los datos miembros que estime oportunos, así como el constructor. Éste debe aceptar como parámetro el número de ciudades con las que se va a trabajar.
2. Defina los métodos que estime oportunos para poder establecer y eliminar conexiones entre dos ciudades cualesquiera, así como para averiguar si dos ciudades están conectadas.
3. Diremos que hay un camino de longitud k entre dos ciudades i y j si existe un conjunto $\{c_1, \dots, c_k\}$ de k ciudades (escalas) de forma que i está conectada con c_1 , c_1 está conectada con c_2 y así sucesivamente hasta c_k la cual está conectada con j .

Construya un método que calcule cuántos caminos de longitud $k = 2$ hay entre dos ciudades, es decir, cuántas formas hay de llegar desde una ciudad origen hasta otra de destino usando dos y sólo dos escalas. Considere que:

- La ciudad de origen no puede ser igual a la ciudad de destino. Por ejemplo, el camino 1→3→1 no sería correcto.
 - Ninguna de las escalas puede ser la propia ciudad de origen o de destino. Por ejemplo, el camino 1→3→5→3 no sería correcto ya que pasa por la misma ciudad de destino (3). Tampoco lo sería el camino 1→5→1→3 ya que pasa por la misma ciudad de origen (1).
 - En cualquiera de los casos anteriores, el método devolverá cero.
4. Podemos representar un camino entre dos ciudades como un objeto de la clase SecuenciaEnteros. que contenga los códigos de las ciudades correspondientes. A dicho objeto lo llamaremos trayecto.
Defina un método de la clase Conexiones para que, dado un trayecto, nos diga si es o no viable, es decir, si están conectadas las ciudades i e $i + 1$ para todas las ciudades i del trayecto.
Por ejemplo, el trayecto $\{1, 3, 2, 5\}$ será viable si y solo si existen las conexiones 1→3, 3→2 y 2→5
 5. Construya un programa principal que haga lo siguiente (supondremos que los datos se introducen correctamente, por lo que no es necesario realizar filtros de entradas de datos):
 - (a) Lea un entero representando el número de ciudades (n)
 - (b) Lea un total de $n \times n$ valores enteros 1 o 0. Conforme se van leyendo los datos, se irán asignando las conexiones correspondientes en un objeto conexiones. Si es un 1, habrá una conexión. Si es un 0, no habrá conexión.
 - (c) El programa leerá dos ciudades origen y destino (enteros entre 0 y $n - 1$) e imprimirá en pantalla el número de caminos con dos escalas que hay desde el origen hasta el destino.
 - (d) A continuación, el programa leerá un conjunto de códigos de ciudades (enteros entre 0 y $n - 1$) hasta que se introduzca el valor -1. Dichos códigos formarán un objeto trayecto de la clase SecuenciaEnteros.
 - (e) El programa dirá si el trayecto es viable o no.

- Ejemplo de entrada:

5

```
0 1 1 1 1
1 0 1 1 0
0 1 0 1 1
0 1 1 0 1
1 1 0 1 0
```

1 4

1 2 4 -1

La salida del programa será, en este caso:

```
Número de conexiones de dos escalas entre las ciudades 1 y 4: 4
El trayecto es viable
```

Puede comprobar manualmente que las conexiones de dos escalas entre las ciudades 1 y 4 son:

```
1 0 2 4
1 0 3 4
1 2 3 4
1 3 2 4
```

- Con el siguiente ejemplo de entrada:

5

```
0 1 0 0 1
1 0 1 1 0
1 1 0 0 1
1 1 1 0 1
1 0 0 1 0
```

1 4

0 4 2 -1

la salida sería:

```
Número de conexiones de dos escalas entre las ciudades 1 y 4: 3
El trayecto no es viable
```