

## Túneles policiales

### Descripción del problema

La policía de una ciudad extremadamente peligrosa ha organizado una red subterránea que les permite moverse rápidamente entre distintos puntos de la ciudad para atender las emergencias que surgen.

Esta red subterránea está formada por estaciones, con salida al exterior, y túneles que unen las distintas estaciones.

En algunas estaciones hay un móvil policial dispuesto a atender cualquier problema que surja en la ciudad. Dado que los túneles son todos exactamente de la misma longitud y que los móviles se mueven a la misma velocidad, el recorrido de cualquier túnel insume una unidad de tiempo.

En la central de policía hay una persona que actúa como coordinador de la red subterránea. Este coordinador es el encargado de recibir los pedidos de ayuda y enviar tres móviles al lugar de la emergencia. Dada la peligrosidad de la ciudad, el coordinador debe asegurarse que los tres móviles arriben al mismo tiempo y lo antes posible al lugar de la emergencia, caso contrario fracasarían en la atención de la misma.

Además, para evitar conflictos de circulación por los túneles, el coordinador debe regular las marchas de los móviles de manera tal que en un mismo instante de tiempo no haya dos móviles en la misma estación (salvo que sea la estación donde ocurre la emergencia). Esto significa que en cada unidad de tiempo un móvil puede o avanzar a otra estación (usando el túnel que las une) o permanecer detenido en una estación a fin de coordinar con sus compañeros.

Con el fin de ayudar al coordinador, se te pide que escribas el programa **TUNELES** en C, C++ o pascal, que teniendo como datos en qué estaciones hay móviles disponibles y en qué estación ocurre la emergencia, determine los tres móviles policiales que deben atender la emergencia, las unidades de tiempo que les insume llegar y las acciones (esperar o avanzar a la próxima estación) que debe realizar cada uno para llegar al lugar del conflicto.

### Aclaraciones

- Siempre existen en la red al menos tres móviles disponibles y al menos tres túneles desembocan en la estación de la emergencia.

- Los tres móviles para atender la emergencia actual se envían cuando los tres móviles de la emergencia anterior dejan de circular por la red.
- Las estaciones se identifican por números consecutivos, empezando desde 1.
- Pueden haber como máximo 120 estaciones, y cada estación puede tener como máximo 20 túneles que la comunican.
- No existen dos túneles distintos que comuniquen el mismo par de estaciones.
- En la estación donde ocurre la emergencia no hay un móvil disponible.
- Los tres móviles deben llegar a la estación donde ocurre la emergencia al mismo tiempo.

### Datos de entrada

Se recibe un archivo **TUNELES.IN** del directorio actual, que contiene:

- Primera línea: tres números separados por blancos indicando: la cantidad  $n$  de estaciones, la cantidad  $m$  de túneles y la cantidad  $K$  de móviles disponibles ( $3 \leq k < n$ ).
- A continuación,  $m$  líneas cada una de ellas conteniendo 2 números separados por blanco que indican los números de estaciones que une un túnel.
- A continuación,  $k$  líneas cada una de ellas conteniendo un número que indica la estación donde hay un móvil disponible.
- Finalmente una línea con un número que indica el número de estación donde ocurre la emergencia.

### Datos de salida

El programa debe generar el archivo **TUNELES.OUT**, en el directorio actual, con:

- Primera línea: un número  $p$  indicando la unidad de tiempo en la que arribarán los tres móviles al lugar de la emergencia, suponiendo que parten en la unidad de tiempo cero.
- Segunda línea: 3 números separados por blanco que indican los 3 números de estaciones donde se encontraban inicialmente los móviles seleccionados.
- A continuación  $p$  líneas, cada una con tres números separados por blanco indicando la acción que tomará cada móvil en cada unidad de tiempo: el número de estación donde permanece en espera o el número de estación a la cual se

mueve. En cada línea, cada uno de los tres números se corresponde en orden con cada móvil dado en la línea 2.

### Ejemplo

En el caso de que el archivo **TUNELES.IN** contenga:

|   |   |   |
|---|---|---|
| 7 | 8 | 4 |
| 1 | 2 |   |
| 2 | 3 |   |
| 6 | 3 |   |
| 3 | 4 |   |
| 3 | 5 |   |
| 1 | 4 |   |
| 6 | 7 |   |
| 5 | 7 |   |
| 1 |   |   |
| 4 |   |   |
| 5 |   |   |
| 7 |   |   |
| 3 |   |   |

El archivo **TUNELES.OUT** podría contener:

|   |   |   |
|---|---|---|
| 2 |   |   |
| 1 | 4 | 7 |
| 2 | 4 | 6 |
| 3 | 3 | 3 |