

Estaciones de Bomberos

En una zona del interior del país donde existen varias poblaciones se necesita situar tres estaciones de bomberos. Cada uno de los camiones de las estaciones de bomberos solo pueden recorrer una cierta cantidad **K** de kilómetros.

Se conocen las distancias que existen entre las poblaciones, puede ser que entre dos de ellas no exista un camino directo pero siempre se podrá encontrar un camino pasando por otras poblaciones.

El problema consiste en encontrar la mejor disposición posible para situar las tres estaciones de bomberos de modo que la cantidad de poblaciones que puedan alcanzar sus camiones sea la mayor.

Note que si una población es alcanzable desde más de una estación a los efectos del problema anterior cuenta como una población (lo que interesa es que la población sea alcanzable desde alguna estación no que tenga que ser alcanzable por todas las estaciones).

Usted debe implementar el método que tenga la siguiente signatura **UbicaEstacionesDeBomberos**:

```
public class CaminoBomberos
{
    /// <summary>
    /// Método que retorna un arreglo con las posiciones de las tres estaciones de
    /// bomberos para que la cantidad de poblaciones protegidas sea la mayor.
    /// </summary>
    /// <param name="distancias">Es una matriz de enteros donde se tienen la distancia
    /// entre todas las poblaciones. La diagonal es 0 ya que la distancia entre de una
    /// población con ella misma es cero. Entre dos poblaciones que no tengan un camino
    /// directo el valor que aparecerá en la tabla es -1.</param>
    /// <param name="alcanzeMaximo">Es la distancia máxima que puede alcanzar cualquiera
    /// de las estaciones de bomberos.</param>
    /// <returns>Array de tres enteros que representan las poblaciones donde se sitúan
    /// las estaciones </returns>
    public static int[] UbicaEstacionesDeBomberos(int[,] distancias, int alcanzeMaximo)
    {
        // Implementar...
    }
}
```

Parámetros:

int[,] distancias : Es una matriz de enteros donde se tienen la distancia entre todas las poblaciones. La diagonal es **0** ya que la distancia entre de una población con ella misma es cero. Entre dos poblaciones que no tengan un camino directo el valor que aparecerá en la tabla es **-1**.

int alcanzeMaximo : Es la distancia máxima que puede alcanzar cualquiera de las estaciones de bomberos.

Retorno:

`int[]` : Array de tres enteros que representan las poblaciones donde se sitúan las estaciones.

Un ejemplo seria para la entrada:

```
int[,] distancias = new int[,]{
    {0,-1,-1,-1,-1,2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    {-1,0,-1,-1,-1,2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,0,-1,-1,2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,0,-1,2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,0,2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
    {2,2,2,2,2,0,-1,-1,-1,-1,-1,-1,2,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,0,-1,-1,-1,-1,-1,2,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,-1,0,-1,-1,-1,-1,2,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,-1,-1,0,-1,-1,-1,2,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,-1,2,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,2,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,2,-1,-1,-1,-1,-1,-1,2},
    {-1,-1,-1,-1,-1,-1,2,2,2,2,2,2,0,-1,-1,-1,-1,-1,-1,-1},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,-1,-1,-1,-1,2},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,-1,-1,-1,2},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,-1,-1,2},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,-1,2},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,-1,2},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,2},
    {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,2,-1,2,2,2,2,2,0}
};
```

Y `alcanzeMaximo = 3` la salida seria `{5,12,20}`, con esta disposición de las estaciones de bomberos se alcanzan todas las poblaciones.