1.- ENUNCIADO DE LA PRÁCTICA

Se trata de calcular el camino de coste mínimo entre cada par de nodos de un grafo dirigido utilizando el algoritmo de Floyd, algoritmo que utiliza el esquema de Programación Dinámica. El resultado del algoritmo será, para cada par de nodos, la secuencia de nodos del camino más corto y su valor total o longitud.

El enunciado del problema, la descripción de la solución, junto con un ejemplo explicativo se puede encontrar en el apartado 5.7 del texto base de la asignatura.

2.- REALIZACIÓN DE LA PRÁCTICA

2.1.- Diseño del algoritmo

La práctica constará de una memoria y de un programa en java original que resuelva el problema aplicando el esquema indicado.

2.2.- Argumentos y parámetros

La práctica se invoca usando la siguiente sintaxis:

```
java  floyd [-t][-h] [fichero entrada] [fichero salida]
```

o

```
java –jar floyd.jar [-t][-h] [fichero entrada] [fichero salida]
```

Los argumentos son los siguientes:

- **-t**: traza cada invocación recursiva de manera que se describa la parametrización de cada llamada recursiva.
- **-h**: muestra una ayuda y la sintaxis del comando.
- **fichero_entrada:** es el nombre del fichero del que se leen los datos, en este caso la representación del grafo.
- **fichero_salida:** es el nombre del fichero que se creará para almacenar la salida. Si el fichero ya existe, el comando dará un error. Si falta este argumento, el programa muestra el resultado por pantalla.

Por ejemplo:

```
$ java floyd -h  <ENTER>

SINTAXIS: floyd [-t][-h] [fichero entrada] [fichero salida]
     -t                   Traza el algoritmo
     -h                   Muestra esta ayuda
     [fichero entrada]    Matriz de adyacencia que representa el grafo
     [fichero salida]     Para cada par de nodos: la lista de nodos del
                          camino más corto y su valor o longitud
```

2.3- Datos de entrada

El formato del fichero de entrada correspondiente a un grafo dirigido de *n* nodos será de *n* filas y cada una contendrá *n* valores separados por espacios. Los valores vacíos se denominarán con un guión (-). Estas *n* filas x *n* columnas representan el grafo mediante una matriz de adyacencia, *M*, en la que el valor *M*[*i,j*] representa el valor de la arista entre los nodos *i* y *j*.

En caso de que el fichero de entrada no exista, leerá los datos por la entrada estándar con similar formato.

2.4- Datos de salida

La salida consistirá para cada par de nodos en:

- Su identificación.
- La lista de nodos del camino más corto y su valor o longitud.

Por ejemplo:

```
 [1, 2]: 3,4,5: 45
…
```

Expresando que el camino más corto entre los nodos 1 y 2 se obtiene pasando por el 3, el 4 y el 5, en ese orden, con un valor, coste o longitud total de 45.

En caso de que el fichero de salida no se indique en la llamada al programa, se escribirá el resultado por la salida estándar.

2.5.- Implementación del algoritmo

El programa se desarrollará en Java siguiendo un diseño orientado a objetos. Los detalles del entorno recomendado se encuentran en la guía de la asignatura. Se valorará el diseño OO y la eficiencia del desarrollo.