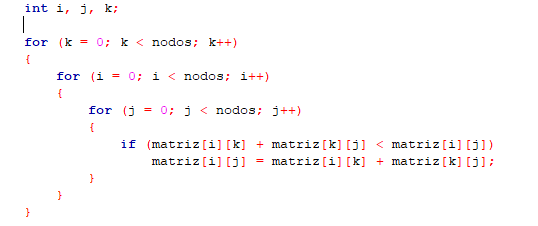
**ALGORITMO DE FLOYD: DISTANCIAS MINIMAS**



Dada una matriz de adyacencia , se realizo un algoritmo que se encarga de comparar todos los posibles caminos entre cada par de vértices. Se produce una iteración “K” sobre cada nodo intermedio, ósea para todo elemento de matriz se prueba si lo mejor para ir de “I” a “J” es a través de un nodo intermedio elegido o como estaba anteriormente, y esto se prueba con todos los nodos de la red. En síntesis, se fija si la distancia entre un nodo I con uno J pasa por un nodo intermedio (seria I, K, J), es mas corta que la distancia directa entre el nodo I y J sin pasar por ningún inntermedio. Si esa ruta tiene una distancia mínima que la directa entre I y J, entonces va a ser la nueva distancia. Asi se va comparando con todos los nodos hasta encontrar las distancias mas cortas.

En el algoritmo, se itera K veces por la cantidad de nodos que haya, se inicializa en 0 porque la posición en una matriz siempre comienza desde 0. Luego se va recorriendo comenzando desde la fila I por cada columna J, obteniendo asi la distancia que hay entre cada uno (siendo I y J ciudades o nodos). Entonces por cada nodo se verifica que peso o distancia se tiene, y luego se evalua si existen caminos o nodos intermedios entre ellos (usando a K como nodo intermedio tanto para I como para J). Si existen estos caminos (de I a K y de K a J), se suman y se verifican si son menor que la distancia entre los nodos a comparar. Si es menor entonces pasa a colocarse ese peso en la matriz de adyacencia.

Tener en cuenta que los nodos o ciudades que no tiene conexión se les puso un numero muy alto (INF = 99999), para cuando se haga la comparación en el if siempre sea una distancia muy alta y por ende no se tome en cuenta. De esta manera se logra recorrer todos y cada uno de los nodos para verificar la existencia de conexiones y distancias minimas en caso de que existan.