

Exersice 108:- Floyd algorithm

Program:-

```
INF = float('inf')
```

```
def floydWarshall(graph):
```

```
    dist = list(map(lambda i: list(map(lambda j: j, i)), graph))
```

```
    V = len(graph)
```

```
        for k in range(V):
```

```
            for i in range(V):
```

```
                for j in range(V):
```

```
                    dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
```

```
    printSolution(dist)
```

```
def printSolution(dist):
```

```
    V = len(dist)
```

```
    print("Following matrix shows the shortest distances between every pair of vertices")
```

```
    for i in range(V):
```

```
        for j in range(V):
```

```
            if dist[i][j] == INF:
```

```
                print("INF", end=" ")
```

```
            else:
```

```
                print(dist[i][j], end=" ")
```

```
    print()
```

```
graph = [[0, 5, INF, 10],
```

```
         [INF, 0, 3, INF],
```

```
         [INF, INF, 0, 1],
```

```
         [INF, INF, INF, 0]]
```

```
floydWarshall(graph)
```

output:-

Following matrix shows the shortest distances between every pair of vertices

0	5	8	9
INF	0	3	4
INF	INF	0	1
INF	INF	INF	0

=== Code Execution Successful ===

Time complexity:- $O(V^2)$