Name: Ganesh K

USN : 2VX23UE015

Expt No.4(A): Floyd's Algorithm.

In [3]:

```
n = 4
inf = float('inf')
graph = [[0, 3, inf, 7],
         [8, 0, 2, inf],
         [5, inf, 0, 1],
         [2, inf, inf, 0]]

def floyd_warshall(graph):
    n = len(graph)
    dist = []
    for row in graph:
        dist.append(row[:])

    for k in range(n):
        for i in range(n):
            for j in range(n):
                dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
    return dist

shortest_paths = floyd_warshall(graph)

print("Shortest paths between all pairs of vertices:")
for row in shortest_paths:
    for dist in row:
        if dist == inf:
            print("inf", end=" ")
        else:
            print(dist, end=" ")

    print()

```

```
Shortest paths between all pairs of vertices:
0 3 5 6
5 0 2 3
3 6 0 1
2 5 7 0
```

In [2]:

```python
n = int(input("Enter the number of Vertices: "))
inf = float('inf')
print("Enter the adjacency matrix (Enter 'inf' for infinity):")
graph = []

for i in range(n):
    row = input().split()
    graph.append([inf if x == 'inf' else int(x) for x in row])

def floyd_warshall(graph):
    n = len(graph)
    dist = []
    for row in graph:
        dist.append(row[:])

    for k in range(n):
        for i in range(n):
            for j in range(n):
                dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
    return dist

shortest_paths = floyd_warshall(graph)

print("Shortest paths between all pairs of vertices:")
for row in shortest_paths:
    for dist in row:
        if dist == inf:
            print("inf", end=" ")
        else:
            print(dist, end=" ")

    print()

```

```
Enter the number of Vertices: 4
Enter the adjacency matrix (Enter 'inf' for infinity):
0 3 inf 7
8 0 2 inf
5 inf 0 1
2 inf inf 0
Shortest paths between all pairs of vertices:
0 3 5 6
5 0 2 3
3 6 0 1
2 5 7 0
```

In [ ]:

```python

```