Exersice 111:- Warshalls algorithm

Program:-

```
def floyd_warshall(graph):
    V = len(graph)
        dist = [[float('inf')] * V for _ in range(V)]
         for i in range(V):
        for j in range(V):
            dist[i][j] = graph[i][j]
        for i in range(V):
        dist[i][i] = 0
        for k in range(V):
        for i in range(V):
            for j in range(V):
                if dist[i][j] > dist[i][k] + dist[k][j]:
                    dist[i][j] = dist[i][k] + dist[k][j]
    return dist
if __name__ == "__main__":
    graph = [
        [0, 3, float('inf'), -4],
        [float('inf'), 0, 7, 1],
        [4, float('inf'), 0, float('inf')],
        [float('inf'), 6, 5, 0]
    ]
     dist = floyd_warshall(graph)
        print("Shortest distances between every pair of vertices:")
    for row in dist:
        print(row)
```

output:-

```
Shortest distances between every pair of vertices:
[0, 2, 1, -4]
[10, 0, 6, 1]
[4, 6, 0, 0]
[9, 6, 5, 0]

=== Code Execution Successful ===
```

Time complexity:-O(V^3)