184. Write a Program to implement Floyd's Algorithm to calculate the shortest paths between all pairs of routers. Simulate a change where the link between Router B and Router D fails. Update the distance matrix accordingly. Display the shortest path from Router A to Router F before and after the link failure.

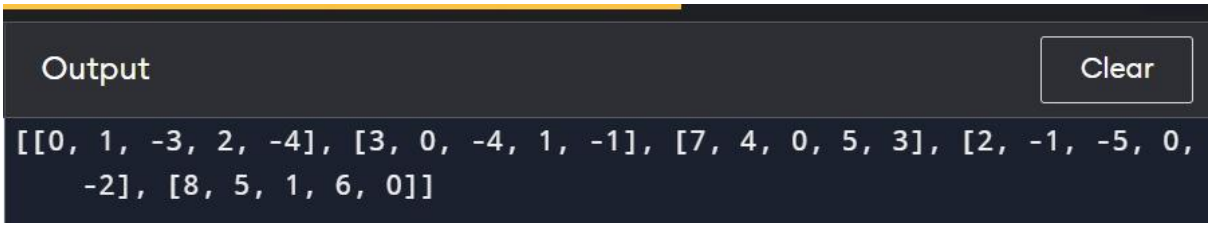
Input as above

Output : Router A to Router F = 5

Program:

```
def floyds_algorithm(graph):
    n = len(graph)
    dist = graph

    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

# Example input graph representing distances between routers
graph = [
    [0, 3, 8, float('inf'), -4],
    [float('inf'), 0, float('inf'), 1, 7],
    [float('inf'), 4, 0, float('inf'), float('inf')],
    [2, float('inf'), -5, 0, float('inf')],
    [float('inf'), float('inf'), float('inf'), 6, 0]
]

# Applying Floyd's Algorithm to calculate shortest paths
distance_matrix = floyds_algorithm(graph)
print(distance_matrix)
```

Output:



Timecomplexity: : O(n^3)