Assignment-15

Implement the Floyd-Warshall algorithm in Java to find the shortest paths between all pairs of vertices in a weighted graph.

```
public class FloydWarshallAlgorithmDemo {

	final static int INF=9999;
	final static int V=4;

	void floydWarshall(int graph[][]) {
		int dist[][]=new int[V][V];
		int i,j,k;

		for(i=0;i<V;i++)
			for(j=0;j<V;j++)
				dist[i][j]=graph[i][j];

		for(k=0;k<V;k++) {
			for(i=0;i<V;i++) {
				for(j=0;j<V;j++) {
					if(dist[i][k]+dist[k][j]<dist[i][j])
						dist[i][j]=dist[i][k]+dist[k][j];
				}//for
			}//for
		}//for
				printSolution(dist);
	}//end of function


	void printSolution(int dist[][]) {
		System.out.println("The following matrix shows the shortest "
				+ "distance between every pair of vertices ");
		for(int i=0;i<V;i++) {
			for(int j=0;j<V;j++) {
				if(dist[i][j]==INF)
```

```
                                System.out.print("INF");
                        else
                                System.out.print(dist[i][j]+" ");
                    }//end for
                    System.out.println();
                }//end for
        }


        public static void main(String[] args) {
                int graph[][]= {{0,10,1,INF},
                                        {10,0,INF,2},
                                        {1,INF,0,1},
                                        {INF,2,1,0}};
                FloydWarshallAlgorithmDemo obj=new
FloydWarshallAlgorithmDemo();
                obj.floydWarshall(graph);

        }

}
```