

```
INF = 99999
```

```
#print the solution matrix
```

```
def printSolution(V, D):  
    print("The following matrix shows the shortest distances between every pair of values")  
    for i in range(V):  
        for j in range(V):  
            if D[i][j] == INF:  
                print("%7s" % "INF", end="")  
            else:  
                print("%7d" % D[i][j], end="")  
        print()
```

```
#Implementing Floyd warshall algorithm
```

```
def floyd(V, C):  
    D = [[0]*V for _ in range(V)]  
  
    for i in range(V):  
        for j in range(V):  
            D[i][j] = C[i][j]  
  
    for k in range(V):  
        for i in range(V):  
            for j in range(V):  
                if D[i][j] > (D[i][k] + D[k][j]):  
                    D[i][j] = D[i][k] + D[k][j]  
  
    printSolution(V, D)
```

```
#Main code
```

```
V = int(input("Enter the number of vertices:"))  
# allocate memory for the cost matrix  
C = [[0]*V for _ in range(V)]  
print("Enter the cost matrix row by row (space-separated): ")  
print("[Enter 99999 for Infinity]")  
print("[Enter 0 for cost (i,i)]")  
for i in range(V):  
    C[i] = list(map(int, input().split()))
```

floyd(V,C)