8. Write a Program to implement Floyd's algorithm and find the lengths of the shortest paths from every pairs of vertices in a given weighted graph

```
INF=9999
def printsolution(V,D):
    print("The All Pair Shortest Path: ")
    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()

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)

V=int(input("Enter the number of vertices:"))
C=[[0]*V for _ in range(V)]
print("Enter the cost matrix:")
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)
```

Output 1

```
Enter the number of vertices:4
Enter the cost matrix:
Enter 99999 for Infinity
Enter 0 for cost(i,i)
    0      10      99999      40
99999      0      99999      20
   50   99999        0      99999
99999  99999       60         0
```

The All Pair Shortest Path:

```
  0   10   90   30
130    0   80   20
 50   60    0   80
110  120   60    0
```

Output 2

Enter the number of vertices:4
Enter the cost matrix:
Enter 99999 for Infinity
Enter 0 for cost(i,i)

```
   0    99999     2      99999
   3      0     99999   99999
99999   5        0         1
   6   99999  99999      0
```

The All Pair Shortest Path:

```
  0   7   2   3
  3   0   5   6
  7   5   0   1
  6  13   8   0
```