**Program 13-Implement All Pair Shortest paths problem using Floyd’s algorithm.**

// C Program for Floyd Warshall Algorithm

#include<stdio.h>

#define V 4

#define INF 99999

void printSolution(int dist[][V]);

void floydWarshall (int graph[][V])

{

int dist[V][V], 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];

}

}

}

printSolution(dist);

}

void printSolution(int dist[][V])

{

printf ("The following matrix shows the shortest distances"

" between every pair of vertices \n");

for (int i = 0; i < V; i++)

{

for (int j = 0; j < V; j++)

{

if (dist[i][j] == INF)

printf("%7s", "INF");

else

printf ("%7d", dist[i][j]);

}

printf("\n");

}

}

int main()

{

int graph[V][V] = { {0,3,INF,7},

{8, 0,2,INF},

{5,INF, 0,1},

{2, INF, INF, 0}

};

floydWarshall(graph);

return 0;

}

**OUTPUT-**

