#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define INF 999

#define V 100

void floydWarshall(int graph[][V], int n) {

int dist[V][V];

int i, j, k;

for (i = 0; i < n; i++)

for (j = 0; j < n; j++)

dist[i][j] = graph[i][j];

for (k = 0; k < n; k++) {

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

if (dist[i][k] != INF && dist[k][j] != INF && dist[i][k] + dist[k][j] < dist[i][j])

dist[i][j] = dist[i][k] + dist[k][j];

}

}

}

printf("Shortest distances between every pair of vertices:\n");

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

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

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

else

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

}

printf("\n");

}

}

int main() {

int n, i, j;

char input[20];

int graph[V][V];

printf("Enter the number of vertices in the graph: ");

scanf("%d", &n);

printf("Enter the adjacency matrix for the graph (Enter INF for infinity):\n");

for (i = 0; i < n; i++) {

for (j = 0; j < n; j++) {

scanf("%s", input);

if (strcmp(input, "INF") == 0)

graph[i][j] = INF;

else

graph[i][j] = atoi(input);

}

}

floydWarshall(graph, n);

return 0;

}