```
// Floyd Warshall Algorithm

#include <iostream>
using namespace std;

// defining the number of vertices
#define nV 4

#define INF 99999

void printMatrix(int A[][nV]);

void floydWarshall(int graph[][nV])
{
  int A[nV][nV], i, j, k;

  for (i = 0; i < nV; i++)
    for (j = 0; j < nV; j++)
      A[i][j] = graph[i][j];

  for (k = 0; k < nV; k++)
  {
    for (i = 0; i < nV; i++)
    {
      for (j = 0; j < nV; j++)
      {
        if (A[i][k] + A[k][j] < A[i][j])
          A[i][j] = A[i][k] + A[k][j];
      }
    }
  }
  printMatrix(A);
}

void printMatrix(int A[][nV])
{
  for (int i = 0; i < nV; i++)
  {
    for (int j = 0; j < nV; j++)
    {
      if (A[i][j] == INF)
        cout << "INF"
           << "  ";
      else
        cout << A[i][j] << "  ";
    }
    cout << endl;
  }
}

int main()
{
  int graph[nV][nV] = {{0, 3, INF, 5},
              {2, 0, INF, 4},
              {INF, 1, 0, INF},
              {INF, INF, 2, 0}};
```

```
  floydWarshall(graph);
}
```