CPP Program to Implement the Floyd-Warshall Algorithm

```
#include <iostream>
using namespace std;

const int INF = 99999;          // Represents Infinity

void floydWarshall (int graph[4], int V)

  int dist[V][V];

for (int i=0; i<V; i++)
   {
    for (int j=0; j<V; j++)
     {
        clist[i][j] = graph[i][j];
     }
   }

// Apply floyd Warshall Algorithm

for (int k=0; k<v; k++)
    {
    for (int i=0; i<V; i++)
      {
     for(int j=0; j<v ; j++)
       {
```

```
            if (dist[i][k] + dist[k][j] < dist[i][j])
            {
                dist[i][j] = dist[i][k] + dist[k][j];
            }
        }
    }
}

cout << "Shortest distances between all pairs
        of Vertices : \n";

for (int i=0; i<V; i++)
{
    for (int j=0; j<V; j++)
    {
        if (dist[i][j] == INF)
        {
            cout << " INF \t";
        }
        else
        {
            cout << dist[i][j] << "\t";
        }
    }
    cout << endl;
}
```

```
int main()
{
    int V = 4;
    int graph[4][4] =
    {
        { 0, 3, INF, 0 },
        { INF, 0, 1, 7 },
        { 5, INF, 0, INF },
        { 2, INF, INF, 0 }
    };

    floydWarshall(graph, V);

    return 0;
}
```


Output

Shortest distances between all pairs of vertices :-

```
0   3   4   0
6   0   1   6
5   8   0   5
2   5   6   0
```