2/7/23 Floyd's Algorithm.

```
#include <stdio.h>
int V;
void PrintSolution(int dist[][V]);
void floydWarshall(int dist[][V]);
{
    int i, j, k;
  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] == 999) {
      Printf("\t999");
  }
  else
    Printf("%7d", dist[i][j]);
  }
Printf("\n");
```

```
    }
}
int main()
{
Printf("Enter the no vertices in the graph:");
Scanf("%d",&v);
    int graph[v][v];
    Printf("Enter the adjacency matrix
    Enter 999 for the infinite edges:\n");
for(int i=0;i<v;i++)
        for(int j=0;j<v;j++)
            Scanf("%d",&graph[i][j]);
floyd warshall(graph);
    return 0;
}
```

Output

```
Enter the no vertices in the graph
    4
Enter the adjacency matrix Enter 999 for the
    infinite edges

    0   1   999  4
   999  0   999  999
    8   2    0   999
   999  6    5    6
```

The following matrix show the shortest dist b/w every pair of vertices.

0	1	9	4
999	0	999	999
8	2	0	12
13	6	5	0

Ex. 7
27/7/23