

## LAB 5

Q) Implement All Pair Shortest paths problem using 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 number 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]);
floydWarshall(graph);
return 0;
}

```

Output:

```

C:\Users\Admin\Desktop\1BM22CS403\ADA\floyds.exe
Enter the number 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 0
The following matrix shows the shortest distances between every pair of vertices
    0      1      9      4
    999      0      999      999
    8      2      0      12
    13      6      5      0
Process returned 0 (0x0)   execution time : 73.077 s
Press any key to continue.

```