LAB 13- Implement
All Pair Shortest paths problem using Floyd's algorithm.

Anitej Prasad

1BM19CS194

4-D

```
#include<stdio.h>
#define V 4
#define INF 1000

void printSolution(int dist[][V]);

int min(int i,int j)
{
  if(i<j)
    return i;
  return j;
}

void floyd(int A[][4])
{
  int i,j,k,P[4][4];
   for(i=0;i<4;i++)
     for(j=0;j<4;j++)
```

```
        P[i][j]=A[i][j];

   for(k=0;k<4;k++)
      for(i=0;i<4;i++)
         for(j=0;j<4;j++)
            P[i][j]=min(P[i][j],P[i][k]+P[k][j]);
   printSolution(P);
}

void printSolution(int dist[][V])   // Print the Shortest pat matrix
{
    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] == INF)
             printf("%7s", "INF");
          else
             printf ("%7d", dist[i][j]);
       }
       printf("\n");
    }
}
```

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