Name - Jay Bhattarai

Subject - ADA LAB

USN - 1BM19IS198.

Implement all pair shortest paths problem using Floyd's algorithm.

→ Solution.

Using C language.

```
#include <stdio.h>
int min (int, int);
void floyds (int p[10] [10], int n)
{
     int i, j, k;
     for (k = 1; k ≤ n; k++)
         for (i = 1; i ≤ n; i++)
             for(j=1; j≤n; j++)
                 if (i==j)
                     p[i][j] = 0;
                 else
                     p[i][j] = min(p[i][j], p[i][k] + p[k][j]);
}
int min (int a, int b)
{    if (a<b)
          return a;
```

```
      else
        return (b);

 }.
int main()
{
      int p[10][10], w,n,e,u,v,i,j;
      printf("\n Number of vertices:");
      scanf ("%d", &n);
      printf ("\n Enter no. of edges: ");
      scanf ("%d", &e);
      for(i=1; i<=n; i++)
      {
           for(j=1; j<=n; j++)
              p[i][j] = 999;
       }
      for(i=1; i<=e; i++)
      {
           printf("\n Enter end vertices of edge with weight",i);
           scanf ("%d %d %d", &u, &v, &w);
             p[u][v] = w;
      }.
      printf ("\n Matrix of input ");
      for (i=1; i<=n; i++)
      {
            for (j=1; j<=n; j++)
                  printf ("%d", p[i][j]);
            printf("\n");
      }.
```

```
floyds(p,n);
printf("\n Transitive Closure:\n");
for(i=1; i≤n; i++)
{
     for (j=1; j≤n; j++)
        printf("%d ", p[i][j]);
     printf("\n");
 }
 printf("\n The shortest paths are: \n");
 for(i=1; i≤n; i++)
      for(j=1; j≤n; j++)
      {
          if(i!=j)
             printf(" \n <%d ;%d >=%d", i,j, p[i][j]);
      }
}
```

The time complexity of this algorithm is,
O(vertices^3).

<u>For eg</u> If no. of vertices is 3.
Time complexity is O(3^3).

Sample Output:

```
Enter number of vertices: 4.
Enter number of edges: 4.
Enter end vertices of edge 1 with its weight.
1 3 2.

Enter the end vertices of edge 2 with its weight.
4 1 2
Enter the end vertices of edge 3 with its weight.
4 2 1.
Enter the end vertices of edge 4 with its weight.
2 3 4.
Matrix of input dada:
999   999   2     999.        [Here, 999 denotes infinite
999   999   4     999                  distance]
999   999   999   999
2     1     999   999
Transitive closure.
0     999   2     999
999   0     4     999
999   999   0     999
2     1     4     0.
The shotest path are:
<1,2>=999              <3,1>= 999
<1,3>= 2               <3,2>= 999
<1,4>=999              <3,4>= 999.
<2,1>= 999             <4,1>= 2
<2,3>= 4               <4,2>= 1
<2,4>=999              <4,3>= 4
```