Name	Siddhesh Sonar
UID no.	2021700063
Experiment No.	9

AIM:	To implement Travelling Salesman Problem
Program	
PROBLEM STATEMENT :	To implement Travelling Salesman Problem
ALGORITHM/ THEORY:	Given a set of cities and distances between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point. Note the difference between Hamiltonian Cycle and TSP. The Hamiltonian cycle problem is to find if there exists a tour that visits every city exactly once. Here we know that Hamiltonian Tour exists (because the graph is complete) and in fact, many such tours exist, the problem is to find a minimum weight Hamiltonian Cycle. 1. Consider city 1 as the starting and ending point. Since the route is cyclic, we can consider any point as a starting point. 2. Generate all (n-1)! permutations of cities. 3. Calculate the cost of every permutation and keep track of the minimum cost permutation. 4. Return the permutation with minimum cost.

PROGRAM:

```c
#include <stdio.h>

int arr[10][10], comp[10], n, cost = 0;

int minimum(int c)
{
    int kmin, min = 999, nc = 999;
    for (int i = 0; i < n; i++)
    {
        if ((arr[c][i] != 0) && (comp[i] == 0))
            if (arr[c][i] + arr[i][c] < min)
            {
                min = arr[i][0] + arr[c][i];
                kmin = arr[c][i];
                nc = i;
            }
    }
    if (min != 999)
        cost += kmin;

    return nc;
}

void findcost(int city)
{
    int i, ncity;

    comp[city] = 1;

    printf("%d->", city + 1);
    ncity = minimum(city);

    if (ncity == 999)
    {
        ncity = 0;
        printf("%d", ncity + 1);
        cost += arr[city][ncity];

        return;
    }

    findcost(ncity);
}
```

```c
int main()
{
    printf("Enter the number of Villages: ");
    scanf("%d", &n);
    // printf("\nCost Matrix\n");
    for (int i = 0; i < n; i++)
    {
        printf("\nEnter Elements of Row %d: ", i + 1);
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &arr[i][j]);
        }

        comp[i] = 0;
    }
    printf("\n\nCost list:");
    for (int i = 0; i < n; i++)
    {
        printf("\n");
        for (int j = 0; j < n; j++)
        {
            printf("\t%d", arr[i][j]);
        }
    }
    printf("\n\nPath:\n");
    findcost(0);
    printf("\n\nMinimum Cost = %d\n ", cost);
    return 0;
}
```

RESULT:

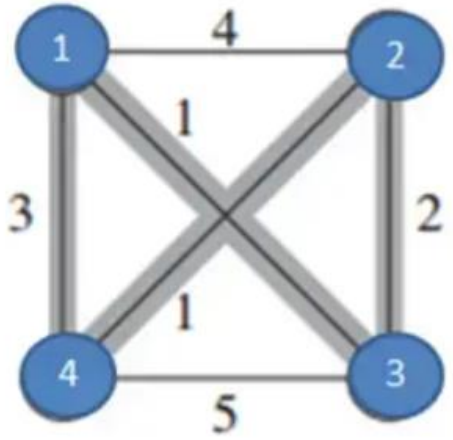
```
Cost list:
        0       4       1       3
        4       0       2       1
        1       2       0       5
        3       1       5       0

Path:
1--->3--->2--->4--->1

Minimum Cost = 7

c:\Siddhesh\Github\DAA\DAA_Exp_9>
```

Example Problem



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 4 & 1 & 3 \\ 4 & 0 & 2 & 1 \\ 1 & 2 & 0 & 5 \\ 3 & 1 & 5 & 0 \end{bmatrix} \end{array}$$

CONCLUSION:	Successfully understood Travelling Salesman Problem and implemented it in C program.