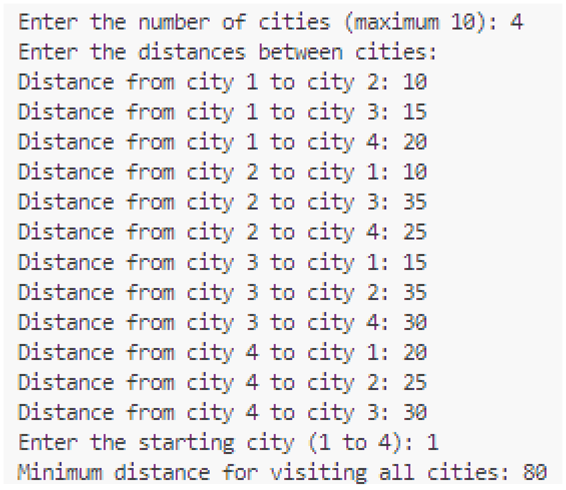
TRAVELLING SALESMAN PROBLEM (TSP)

```
#include <stdio.h>

#include <limits.h>

#define MAX_CITIES 10


void inputDistances(int distances[MAX_CITIES][MAX_CITIES], int numCities)

{

    int i, j;

    for (i = 0; i < numCities; i++)

    {

        for (j = 0; j < numCities; j++)

        {

            if (i != j)

            {

                printf("Distance from city %d to city %d: ", i + 1, j + 1);

                scanf("%d", &distances[i][j]);

            }

            else

            {

                distances[i][j] = 0;

            }

        }

    }

}


int tsp(int distances[MAX_CITIES][MAX_CITIES], int numCities, int currentCity, int
visitedCities)

{

    if (visitedCities == (1 << numCities) - 1)

    {
        return distances[currentCity][0];
    }
    int minDistance = INT_MAX;
    for (int nextCity = 0; nextCity < numCities; nextCity++)
    {
        if (!(visitedCities & (1 << nextCity)))
        {
            int distance = distances[currentCity][nextCity] +
                           tsp(distances, numCities, nextCity, visitedCities | (1 << nextCity));
            if (distance < minDistance)
            {
                minDistance = distance;
            }
        }
    }
    return minDistance;
}

int main()
{
    int numCities, distances[MAX_CITIES][MAX_CITIES], i, j;
    printf("Enter the number of cities (maximum %d): ", MAX_CITIES);
    scanf("%d", &numCities);
    if (numCities <= 0 || numCities > MAX_CITIES)
    {
        printf("Invalid number of cities. Please enter a number between 1 and %d.\n",
MAX_CITIES);
        return 1;
    }
    printf("Enter the distances between cities:\n");
    inputDistances(distances, numCities);
    printf("Enter the starting city (1 to %d): ", numCities);
    int startCity;
    scanf("%d", &startCity);
    startCity--;
    if (startCity < 0 || startCity >= numCities)
    {
        printf("Invalid starting city. Please enter a number between 1 and %d.\n", numCities);
        return 1;
    }
    int minDistance = tsp(distances, numCities, startCity, 1 << startCity);
    printf("Minimum distance for visiting all cities: %d\n", minDistance);
    return 0;
}
```

OUTPUT



```
Enter the number of cities (maximum 10): 4
Enter the distances between cities:
Distance from city 1 to city 2: 10
Distance from city 1 to city 3: 15
Distance from city 1 to city 4: 20
Distance from city 2 to city 1: 10
Distance from city 2 to city 3: 35
Distance from city 2 to city 4: 25
Distance from city 3 to city 1: 15
Distance from city 3 to city 2: 35
Distance from city 3 to city 4: 30
Distance from city 4 to city 1: 20
Distance from city 4 to city 2: 25
Distance from city 4 to city 3: 30
Enter the starting city (1 to 4): 1
Minimum distance for visiting all cities: 80
```