```
//C program to implement travelling
saleperson problem
#include <stdio.h>
#include <stdlib.h>

#define N 4 // Number of cities

int city[N][N]; // Cost matrix

int minCost = 9999;
int bestPath[N];

// Function to calculate the cost of a given
path
int calculateCost(int path[])
{
    int i;
    int cost = 0;
    for (i = 0; i < N - 1; i++)
    {
        cost += city[path[i]][path[i + 1]];
```

```
    }
    cost += city[path[N - 1]][path[0]]; //
Return to the starting city
    return cost;
}

// Function to generate permutations of a
given path
void generatePermutations(int path[], int
start, int end)
{
    int temp,i;
    if (start == end)
    {
        int cost = calculateCost(path);
        if (cost < minCost)
        {
            minCost = cost;
            for (i = 0; i < N; i++)
            {
                bestPath[i] = path[i];
```

```
            }
        }
    }
    else
    {
        for (i = start; i <= end; i++)
        {
            temp = path[start];
            path[start] = path[i];
            path[i] = temp;

            generatePermutations(path, start +
1, end);

            temp = path[start];
            path[start] = path[i];
            path[i] = temp;
        }
    }
}
```

```
int main()
{
    int path[N],i,j;

    // Initialize the cost matrix
    printf("Enter the cost matrix:\n");
    for (i = 0; i < N; i++)
    {
        for (j = 0; j < N; j++)
        {
            scanf("%d", &city[i][j]);
        }
    }

    // Initialize the path array
    for (i = 0; i < N; i++)
    {
        path[i] = i;
    }

    // Generate permutations and calculate
```

```
costs
    generatePermutations(path, 1, N - 1);

    // Print the results
    printf("Minimum cost: %d\n", minCost);
    printf("Best path: ");
    for (i = 0; i < N; i++)
    {
        printf("%d ", bestPath[i]);
    }
    printf("%d\n", bestPath[0]); // Return to
the starting city

    return 0;
}
```