Travelling sales man

```c
#include <stdio.h>
#include <limits.h>

#define MAX 10

int dist[MAX][MAX], n;
int minCost = INT_MAX;
int bestPath[MAX];

void calculateCost(int path[]) {
    int cost = 0;

    for (int i = 0; i < n - 1; i++) {
        cost += dist[path[i]][path[i + 1]];
    }

    // Adding the cost to return to the starting city
    cost += dist[path[n - 1]][path[0]];

    if (cost < minCost) {
        minCost = cost;

        // Store the best path
        for (int i = 0; i < n; i++) {
            bestPath[i] = path[i];
        }
    }
}

void permute(int path[], int left, int right) {
    if (left == right) {
        calculateCost(path);
        return;
    }

    for (int i = left; i <= right; i++) {
        // Swap the elements to generate permutations
        int temp = path[left];
        path[left] = path[i];
        path[i] = temp;
```

```
        permute(path, left + 1, right);

        // Backtrack by swapping again
        temp = path[left];
        path[left] = path[i];
        path[i] = temp;
    }
}

int main() {
    printf("Enter the number of cities: ");
    scanf("%d", &n);

    printf("Enter the distance matrix:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            scanf("%d", &dist[i][j]);
        }
    }

    int path[n];
    for (int i = 0; i < n; i++) {
        path[i] = i;
    }

    permute(path, 0, n - 1);

    printf("Minimum cost to visit all cities: %d\n", minCost);
    printf("The best path is: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", bestPath[i]);
    }
    printf("\n");

    return 0;
}
```

```
Enter the number of cities: 4
Enter the distance matrix:
0 10 15 20
10 0 35 25
15 35 0 30
20 25 30 0
```

```
Minimum cost to visit all cities: 80
The best path is: 0 1 3 2
```