Practical no: 6

Name: Saloni Vishwakarma
Batch-Roll no: C1-13
Subject: DAA Lab
Date of execution: 24 June 2023

Aim: To implement Travelling Salesman algorithm using recursion in C.

Travelling Salesman Algorithm (Code and Output):

```
#include <stdio.h>
#define N 4 // Number of cities
void swap(int *p, int *q) {
  int temp = *p;
  *p = *q;
  *q= temp;
}

int calCost(int path[], int mat[N][N]) {
   int cost = 0, i;
   for ( i = 0; i < N - 1; i++) {
      cost += mat[path[i]][path[i+1]];
   }
   cost += mat[path[N-1]][path[0]]; // Return to the starting city
   return cost;
}

void permutation(int path[], int mat[N][N], int start, int end,int *minCost, int *op) {
   int i;
   if (start == end) {
      int cost = calCost(path, mat);
      if (cost < *minCost) {
         *minCost = cost;
         for ( i = 0; i < N; i++) {
            op[i] = path[i];
         }
```

```
        }
    }
    else
    {
        for ( i = start; i <= end; i++) {
            swap(&path[start], &path[i]);
            permutation(path, mat, start + 1, end, minCost,op);
            swap(&path[start], &path[i]); // Backtrack
        }
    }
}

void printPath(int path[]) {
    int i;
    for ( i = 0; i < N; i++) {
        printf("%c ->", 'A' + path[i]);
    }
    printf("A\n"); // Return to the starting city
}

int main()
{
    int i, j, mat[N][N], path[N], op[N], minCost = 99999; // Initialize with a large value
    // Get distance matrix from the user
    printf("\n *****Travelling Salesman******\n");
    printf("\n Enter the adjacency matrix:\n");
    for ( i = 0; i < N; i++) {
        for ( j = 0; j < N; j++) {
            if(j==0)
            printf(" ");
            scanf("%d", &mat[i][j]);
        }
    }
    // Initialize the path array
    for ( i = 0; i < N; i++) {
        path[i] = i;
    }
    // Generate all permutations of the cities and find the optimal path
```

```
    permutation(path, mat, 0, N-1, &minCost, op);
    // Print the optimal path and cost
    printf("\n Optimal Path: ");
    printPath(op);
    printf("\n Cost: %d\n", minCost);
    return 0;
}
```

```
*****Travelling Salesman******

Enter the adjacency matrix:
0 4 1 3
4 0 2 1
1 2 0 5
3 1 5 0

Optimal Path: A ->C ->B ->D ->A

Cost: 7


...Program finished with exit code 0
Press ENTER to exit console.
```

Conclusion: We have successfully studied and implemented Travelling Salesman algorithm using recursion in C.