Aim: Implement Travelling Salesman Problem using nearest neighbor classifier method

Name: Ghanishtha Anil Rane

Roll No:3363

Class: TEIT

```
import sys
def nearest_neighbor(curr, unvisited, dist_matrix):
    nearest = sys.maxsize
    neighbor = None
    for city in unvisited:
        if dist_matrix[curr][city] < nearest:
            nearest = dist_matrix[curr][city]
            neighbor = city
    return neighbor, nearest


def tsp_nn(dist_matrix):
    n = len(dist_matrix)
    tour = [0] * 5
    unvisited = set(range(1, n))
    curr_city = 0


    for i in range(1, n):
        next_city, dist = nearest_neighbor(curr_city, unvisited, dist_matrix)
        tour[i] = next_city
        curr_city = next_city
        unvisited.remove(next_city)


    tour[0]=0
    cost=sum(dist_matrix[tour[i]][tour[i+1]] for i in range(n-1))
```

```
    cost+=dist_matrix[tour[n-1]][tour[0]]
    return tour,cost
import numpy as np

Rows = int(input("Give the number of rows:"))
Columns = int(input("Give the number of columns:"))

print("Please write the elements of the matrix in a single line and separated by a space: ")

# User will give the entries in a single line
elements = list(map(int, input().split()))

# Printing the matrix given by the user
dist_matrix = np.array(elements).reshape(Rows, Columns)
print(dist_matrix)

tour, cost = tsp_nn(dist_matrix)
print("Tour:", tour)
print("Total cost:", cost)
```

OUTPUT:

```
Give the number of rows:4
Give the number of columns:4
Please write the elements of the matrix in a single line and separated by a space:
0 5 15 4 5 0 35 25 15 35 0 30 4 25 30 0
[[ 0  5 15  4]
 [ 5  0 35 25]
 [15 35  0 30]
 [ 4 25 30  0]]
Tour: [0, 3, 1, 2, 0]
Total cost: 79
```