```python
import sys

def nearest_neighbor_tsp(distances):
    num_cities = len(distances)

    # Start from the first city (arbitrary choice)
    tour = [0]  # Store the tour as a list of city indices
    visited = set([0])  # Track visited cities

    current_city = 0
    total_distance = 0

    while len(visited) < num_cities:
        nearest_city = None
        min_distance = sys.maxsize

        # Find the nearest unvisited city
        for next_city in range(num_cities):
            if next_city not in visited and distances[current_city][next_city] < min_distance:
                nearest_city = next_city
                min_distance = distances[current_city][next_city]

        # Move to the nearest city
        tour.append(nearest_city)
        visited.add(nearest_city)
        total_distance += min_distance
        current_city = nearest_city

    # Complete the tour by returning to the starting city
    tour.append(0)
    total_distance += distances[current_city][0]

    return tour, total_distance

# Example usage:
if __name__ == "__main__":
    # Example distance matrix (symmetric, square matrix)
#    distances = [[0, 10, 15, 20], [10, 0, 35, 25], [15, 35, 0, 30], [20, 25, 30, 0]]
    distances = [[ 0,  4,  8,  9, 12], [ 4,  0,  6,  8,  9], [ 8,  6,  0, 10, 11], [ 9,  8, 10,  0,  7], [12,  9, 11,  7,  0]]
    # Run nearest neighbor TSP algorithm
    tour, total_distance = nearest_neighbor_tsp(distances)

    # Print the tour and total distance
    print("Nearest Neighbor TSP Tour:", tour)
    print("Total Distance:", total_distance)
```