AIM:

The aim is to solve the salesman travelling problem using python program.

PROGRAM:

from itertools import permutations

def calculate\_total\_distance(order, distances):

total\_distance = 0

for i in range(len(order) - 1):

total\_distance += distances[order[i]][order[i+1]]

total\_distance += distances[order[-1]][order[0]] # Return to the starting city

return total\_distance

def traveling\_salesman\_bruteforce(distances):

cities = len(distances)

if cities < 2:

print("Not enough cities for the problem.")

return

# Generate all possible orderings of cities

all\_permutations = permutations(range(cities))

min\_distance = float('inf')

optimal\_order = None

for order in all\_permutations:

total\_distance = calculate\_total\_distance(order, distances)

if total\_distance < min\_distance:

min\_distance = total\_distance

optimal\_order = order

return optimal\_order, min\_distance

if \_\_name\_\_ == "\_\_main\_\_":

# Example distances between cities (symmetric matrix)

example\_distances = [

[0, 10, 15, 20],

[10, 0, 35, 25],

[15, 35, 0, 30],

[20, 25, 30, 0]

]

optimal\_order, min\_distance = traveling\_salesman\_bruteforce(example\_distances)

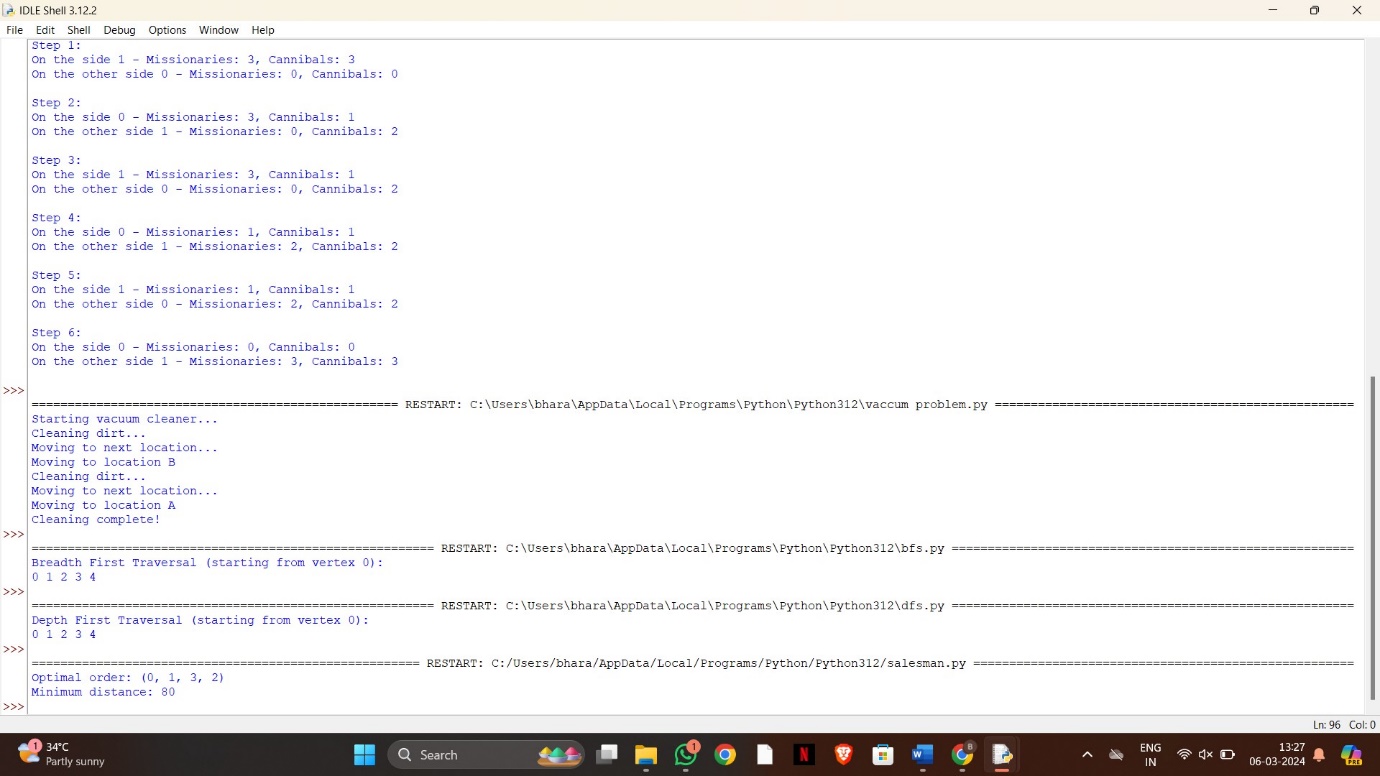
print("Optimal order:", optimal\_order)

print("Minimum distance:", min\_distance)

INPUT:

Optimal order:(0,1,3,2)

OUTPUT:



RESULT:

Thus the solving of salesman traveling using python program is successfully completed.