**TASK 5 :**

**To Implement Ant Colony Optimization to Optimize Ride-Sharing Trip Duration using Python.**

**Program :**

import numpy as np

from numpy import inf

d = np.array([[0, 7, 9, 14, 0, 0],

[7, 0, 10, 15, 0, 0],

[9, 10, 0, 11, 0, 2],

[14, 15, 11, 0, 9, 0],

[0, 0, 0, 9, 0, 6],

[0, 0, 2, 0, 6, 0]])

iteration = 100

n\_ants = 6

n\_citys = 6

m = n\_ants

n = n\_citys

e = 0.5

alpha = 1

beta = 2

visibility = 1 / d

visibility[visibility == inf] = 0

pheromne = 0.1 \* np.ones((n, n))

rute = np.ones((m, n + 1))

for ite in range(iteration):

rute[:, 0] = 1

for i in range(m):

temp\_visibility = np.array(visibility)

for j in range(n - 1):

cur\_loc = int(rute[i, j] - 1)

temp\_visibility[:, cur\_loc] = 0

p\_feature = np.power(pheromne[cur\_loc, :], alpha)

v\_feature = np.power(temp\_visibility[cur\_loc, :], beta)

combine\_feature = p\_feature \* v\_feature

total = np.sum(combine\_feature)

if total == 0:

probs = np.ones(n) / n

else:

probs = combine\_feature / total

cum\_prob = np.cumsum(probs)

r = np.random.random\_sample()

city\_idx = np.nonzero(cum\_prob > r)[0]

if len(city\_idx) == 0: # fallback to random city

city = np.random.randint(1, n + 1)

else:

city = city\_idx[0] + 1

rute[i, j + 1] = city

left = list(set(range(1, n + 1)) - set(rute[i, :-2]))[0]

rute[i, -2] = left

rute\_opt = np.array(rute)

dist\_cost = np.zeros((m, 1))

for i in range(m):

s = 0

for j in range(n - 1):

s += d[int(rute\_opt[i, j]) - 1, int(rute\_opt[i, j + 1]) - 1]

dist\_cost[i] = s

dist\_min\_loc = np.argmin(dist\_cost)

dist\_min\_cost = dist\_cost[dist\_min\_loc][0]

best\_route = rute[dist\_min\_loc, :]

pheromne = (1 - e) \* pheromne

for i in range(m):

for j in range(n - 1):

dt = 1 / (dist\_cost[i] + 1e-6) # avoid div by zero

pheromne[int(rute\_opt[i, j]) - 1, int(rute\_opt[i, j + 1]) - 1] += dt

print("route of all the ants at the end :")

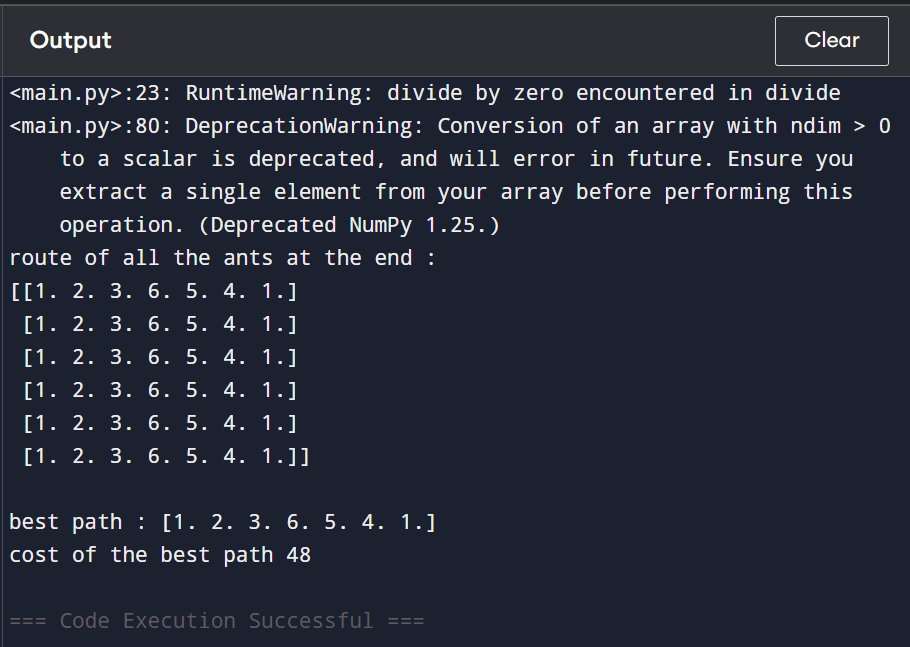
print(rute\_opt)

print()

print("best path :", best\_route)

print("cost of the best path", int(dist\_min\_cost) + d[int(best\_route[-2]) - 1, 0])

**Output :**

****