**Practicle-6 AI Date:- 04-08-2025**

**Q1)8 puzzle problem**

**Code:-**

from collections import deque

goal = "123456780"

moves ={

0: [1,3],

1: [0,2,4],

2: [1,5],

3: [0,4,6],

4: [1,3,5,7],

5: [2,4,8],

6: [3,7],

7: [4,6,8],

8: [5,7]

}

def bfs(start):

visited = set()

queue = deque([(start, [])])

while queue:

state, path = queue.popleft()

if state == goal:

return path + [state]

if state in visited:

continue

visited.add(state)

zero = state.index('0')

for move in moves[zero]:

new\_state = list(state)

new\_state[zero], new\_state[move] = new\_state[move], new\_state[zero]

queue.append((''.join(new\_state), path + [state]))

return None

start = "123405678"

solution = bfs(start)

if solution:

print("Steps to solve:")

for s in solution:

print(s[0:3])

print(s[3:6])

print(s[6:9])

print("------")

else:

print("No solution found")

**O/P:**

**Steps to solve:**

**123**

**405**

**678**

**------**

**123**

**450**

**678**

**------**

**123**

**458**

**670**

**------**

**123**

**458**

**607**

**------**

**123**

**458**

**067**

**------**

**123**

**058**

**467**

**------**

**123**

**508**

**467**

**------**

**123**

**568**

**407**

**------**

**123**

**568**

**470**

**------**

**123**

**560**

**478**

**------**

**123**

**506**

**478**

**------**

**123**

**056**

**478**

**------**

**123**

**456**

**078**

**------**

**123**

**456**

**708**

**------**

**123**

**456**

**780**

**Q2) Missionaries and Cannibals**

moves = [(2,0),(0,2),(1,1),(1,0),(0,1)]

def is\_valid(m\_left,c\_left,m\_right,c\_right):

if m\_left < 0 or c\_left< 0 or m\_right < 0 or c\_right < 0:

return False

if (m\_left > 0 and m\_left < c\_left) or (m\_right > 0 and m\_right < c\_right):

return False

return True

def solve():

from collections import deque

start = (3,3,1)

goal = (0,0,0)

queue = deque()

queue.append((start, [start]))

visited = set()

while queue:

(m\_left, c\_left, boat), path = queue.popleft()

if (m\_left, c\_left, boat) in visited:

continue

visited.add((m\_left, c\_left, boat))

if (m\_left, c\_left, boat) == goal:

return path

for m,c in moves:

if boat == 1:

new\_m\_left = m\_left - m

new\_c\_left = c\_left - c

new\_boat = 0

else:

new\_m\_left = m\_left + m

new\_c\_left = c\_left + c

new\_boat = 1

new\_m\_right = 3 - new\_m\_left

new\_c\_right = 3 - new\_c\_left

if is\_valid(new\_m\_left, new\_c\_left, new\_m\_right, new\_c\_right):

new\_state = (new\_m\_left, new\_c\_left, new\_boat)

if new\_state not in visited:

queue.append((new\_state, path + [new\_state]))

return None

steps = solve()

if steps:

for i, (m,c,b) in enumerate(steps):

side = "Left" if b == 1 else "Right"

print(f"Step {i} : Missionaries Left : {m}, Cannibals Left: {c}, Boat on {side} side")

else:

print("No solution found")

**O/P:**

**Step 0 : Missionaries Left : 3, Cannibals Left: 3, Boat on Left side**

**Step 1 : Missionaries Left : 3, Cannibals Left: 1, Boat on Right side**

**Step 2 : Missionaries Left : 3, Cannibals Left: 2, Boat on Left side**

**Step 3 : Missionaries Left : 3, Cannibals Left: 0, Boat on Right side**

**Step 4 : Missionaries Left : 3, Cannibals Left: 1, Boat on Left side**

**Step 5 : Missionaries Left : 1, Cannibals Left: 1, Boat on Right side**

**Step 6 : Missionaries Left : 2, Cannibals Left: 2, Boat on Left side**

**Step 7 : Missionaries Left : 0, Cannibals Left: 2, Boat on Right side**

**Step 8 : Missionaries Left : 0, Cannibals Left: 3, Boat on Left side**

**Step 9 : Missionaries Left : 0, Cannibals Left: 1, Boat on Right side**

**Step 10 : Missionaries Left : 1, Cannibals Left: 1, Boat on Left side**

**Step 11 : Missionaries Left : 0, Cannibals Left: 0, Boat on Right side**