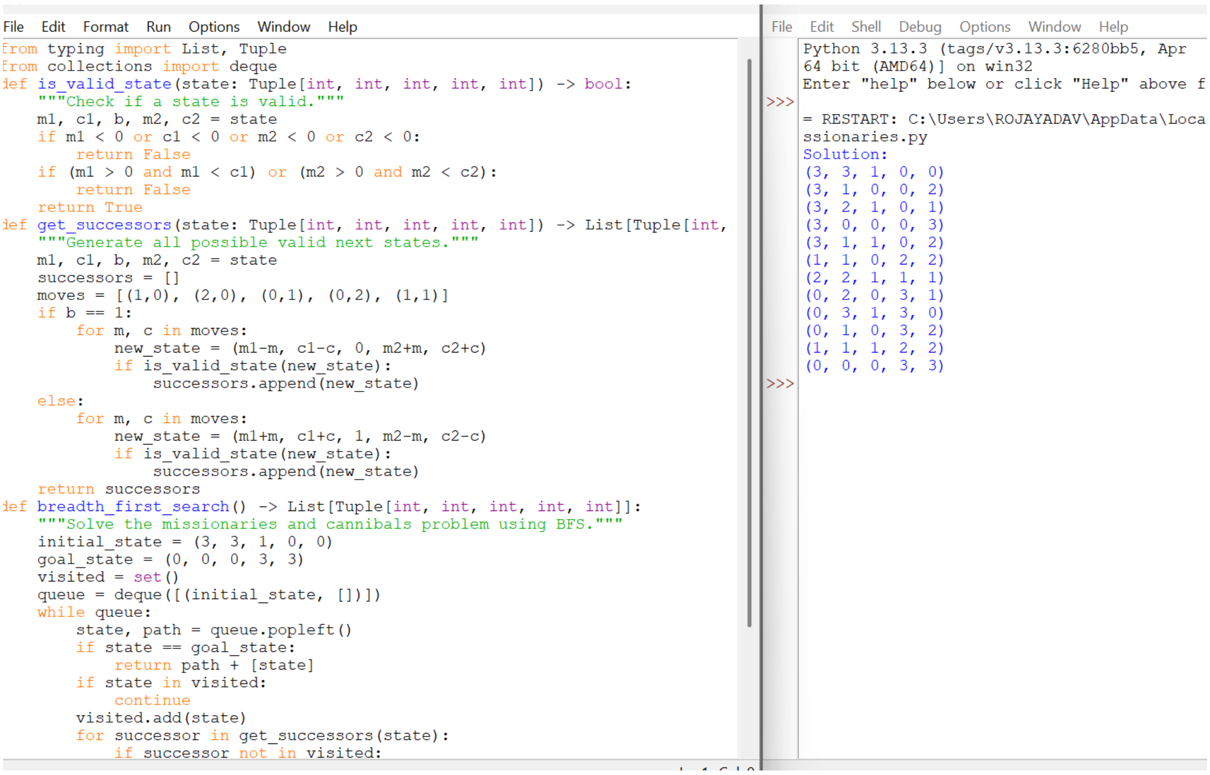
Write the python program for Missionaries Cannibal problem.


File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

```python
from typing import List, Tuple
from collections import deque
def is_valid_state(state: Tuple[int, int, int, int, int]) -> bool:
    """Check if a state is valid."""
    m1, c1, b, m2, c2 = state
    if m1 < 0 or c1 < 0 or m2 < 0 or c2 < 0:
        return False
    if (m1 > 0 and m1 < c1) or (m2 > 0 and m2 < c2):
        return False
    return True
def get_successors(state: Tuple[int, int, int, int, int]) -> List[Tuple[int,
    """Generate all possible valid next states."""
    m1, c1, b, m2, c2 = state
    successors = []
    moves = [(1,0), (2,0), (0,1), (0,2), (1,1)]
    if b == 1:
        for m, c in moves:
            new_state = (m1-m, c1-c, 0, m2+m, c2+c)
            if is_valid_state(new_state):
                successors.append(new_state)
    else:
        for m, c in moves:
            new_state = (m1+m, c1+c, 1, m2-m, c2-c)
            if is_valid_state(new_state):
                successors.append(new_state)
    return successors
def breadth_first_search() -> List[Tuple[int, int, int, int, int]]:
    """Solve the missionaries and cannibals problem using BFS."""
    initial_state = (3, 3, 1, 0, 0)
    goal_state = (0, 0, 0, 3, 3)
    visited = set()
    queue = deque([(initial_state, [])])
    while queue:
        state, path = queue.popleft()
        if state == goal_state:
            return path + [state]
        if state in visited:
            continue
        visited.add(state)
        for successor in get_successors(state):
            if successor not in visited:
```

```
Python 3.13.3 (tags/v3.13.3:6280bb5, Apr
64 bit (AMD64)] on win32
Enter "help" below or click "Help" above f
>>>
= RESTART: C:\Users\ROJAYADAV\AppData\Loca
ssionaries.py
Solution:
(3, 3, 1, 0, 0)
(3, 1, 0, 0, 2)
(3, 2, 1, 0, 1)
(3, 0, 0, 0, 3)
(3, 1, 1, 0, 2)
(1, 1, 0, 2, 2)
(2, 2, 1, 1, 1)
(0, 2, 0, 3, 1)
(0, 3, 1, 3, 0)
(0, 1, 0, 3, 2)
(1, 1, 1, 2, 2)
(0, 0, 0, 3, 3)
>>>
```