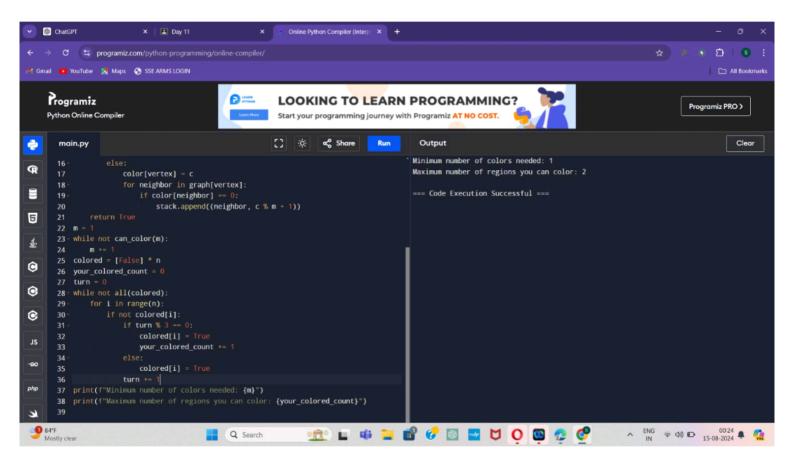
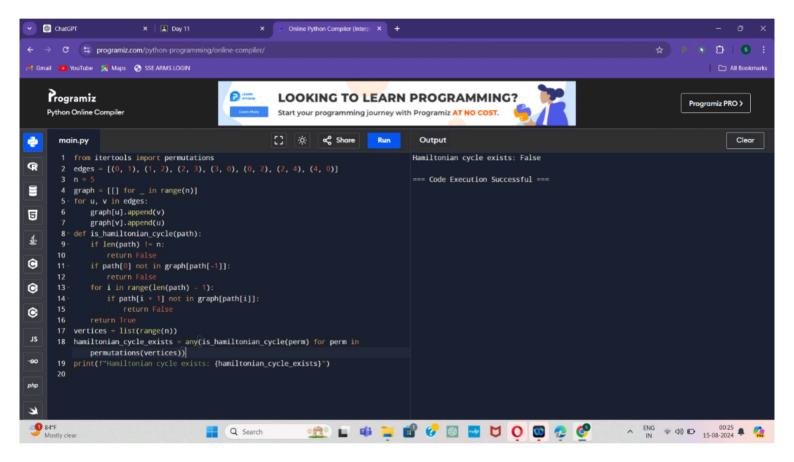
```
are your programming journey mail regramment in over.
                                                                                                               main.py
                                                          -ò-
                                                                ∝ Share
                                                                             Run
                                                                                        Output
                                                                                      Maximum number of regions you can color: 2
1 edges = [(0, 1), (1, 2), (2, 3), (3, 0), (0, 2)]
   graph = [[] for _ in range(n)]
for u, v in edges:
                                                                                       === Code Execution Successful ===
       graph[u].append(v)
       graph[v].append(u)
6
       color = [0] * n
       stack = [(0, 1)]
       valid = True
       while stack:
           vertex, c = stack.pop()
           if color[vertex] != 0:
                if color[vertex] != c:
                   valid = False
16
                   break
               color[vertex] = c
               for neighbor in graph[vertex]:
20
                   if color[neighbor] == 0:
                       stack.append((neighbor, c % m + 1))
        if valid:
```

```
edges = [(0, 1), (1, 2), (2, 3), (3, 0), (0, 2)]
                                                                                        Minimum number of colors needed: 1
                                                                                        Maximum number of regions you can color: 2
   graph = [[] for _ in range(n)]
                                                                                         === Code Execution Successful ===
   for u, v in edges:
       graph[u].append(v)
       graph[v].append(u)
   def can_color(m):
       color = [0] * n
stack = [(0, 1)]
       while stack:
           vertex, c = stack.pop()
            if color[vertex] != 0:
                if color[vertex] != c:
                color[vertex] = c
                for neighbor in graph[vertex]:
                    if color[neighbor] == 0:
                        stack.append((neighbor, c % m + 1))
22
   while not can_color(m):
       m += 1
    colored = [False] * r
```





```
ain.py
                                                        -<u>`</u>ó.
                                                              ∝ Share
 edges = [(0, 1), (1, 2), (2, 3), (3, 0), (0, 2)]
                                                                                    Hamiltonian cycle exists: True
 graph = [[] for _ in range(n)]
                                                                                     === Code Execution Successful ===
 for u, v in edges:
     graph[u].append(v)
     graph[v].append(u)
 path = [-1] * n
 path[0] = 0
 def can_find_hamiltonian_cycle(pos):
     if pos == n:
         return path[0] in graph[path[-1]]
     for v in range(1, n):
         if v not in path and path[pos - 1] in graph[v]:
             path[pos] = v
             if can_find_hamiltonian_cycle(pos + 1):
             path[pos] = -1
 hamiltonian_cycle_exists = can_find_hamiltonian_cycle(1)
 print(f"Hamiltonian cycle exists: {hamiltonian_cycle_exists}")
```

