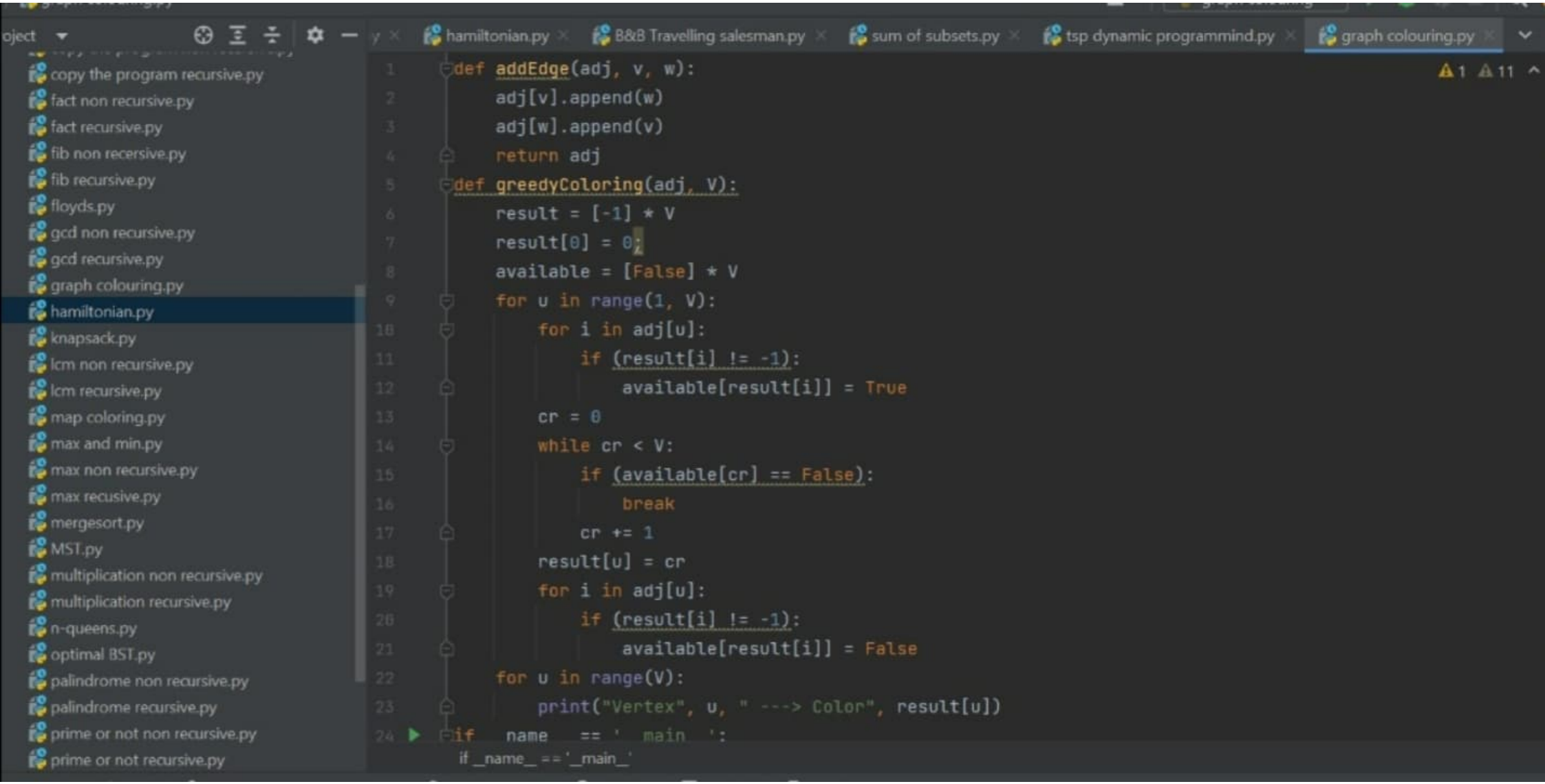
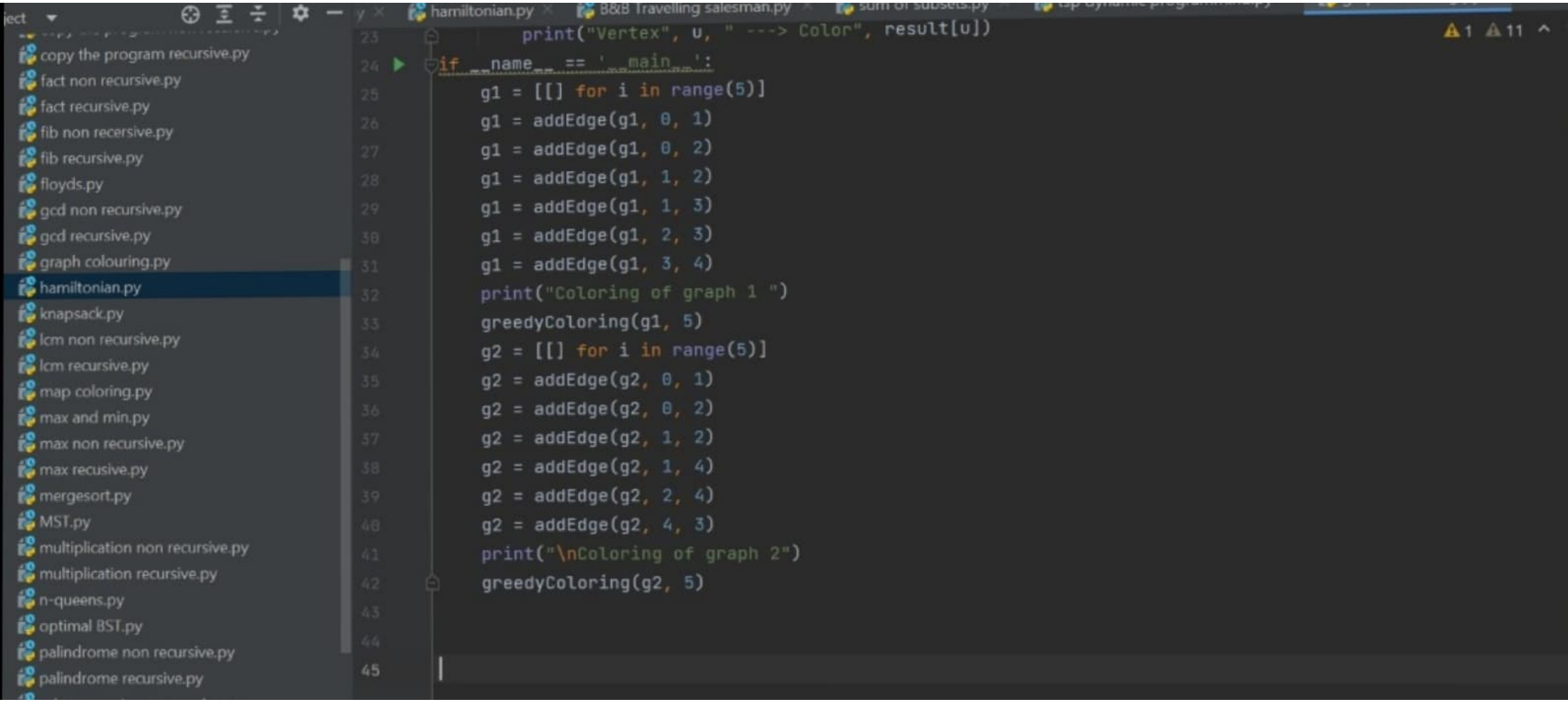
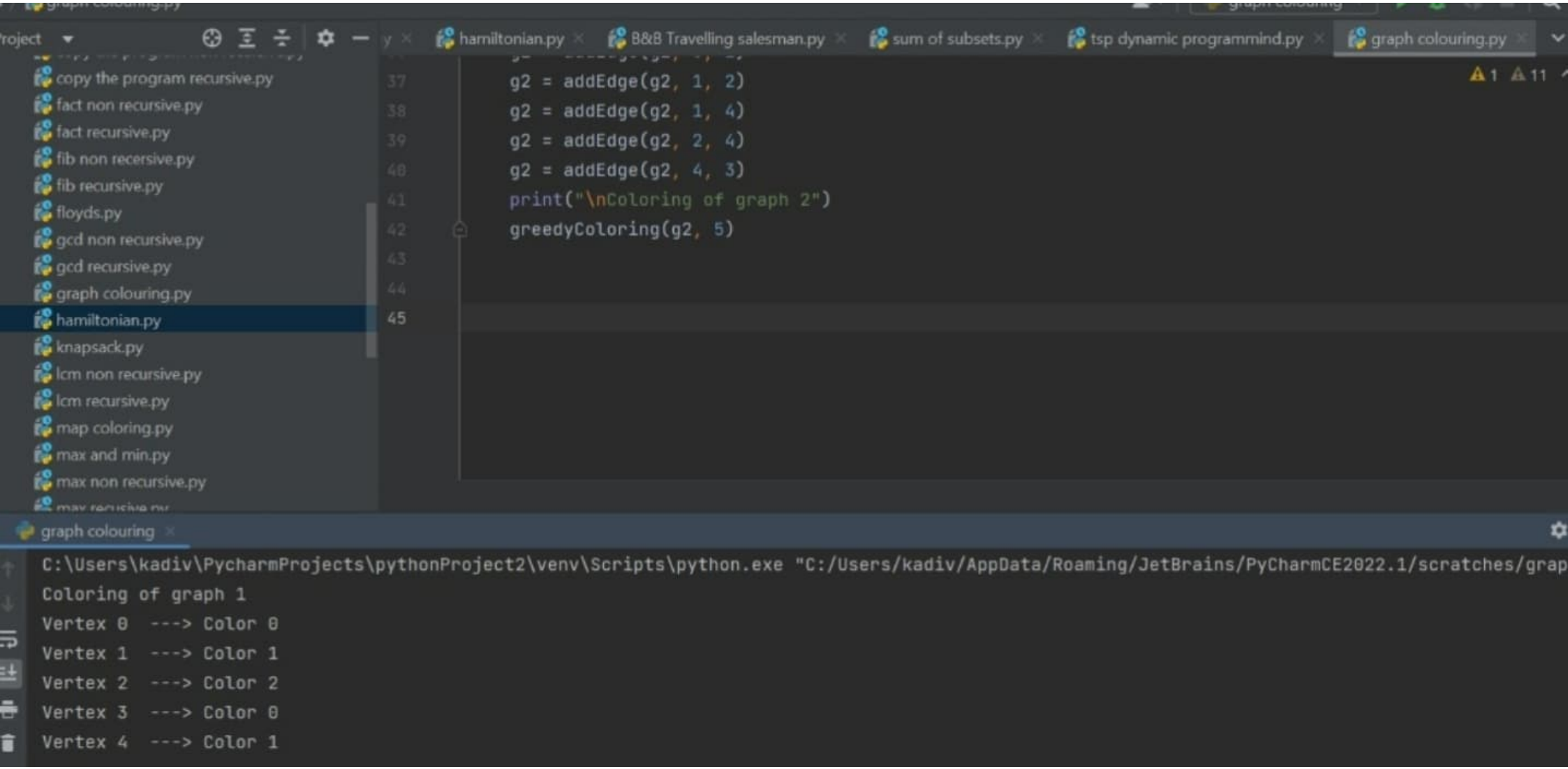

hamiltonian.py
B&B Travelling salesman.py
sum of subsets.py
tsp dynamic programmind.py
graph colouring.py
copy the program recursive.py
fact non recursive.py
fact recursive.py
fib non recersive.py
fib recursive.py
floyds.py
gcd non recursive.py
gcd recursive.py
graph colouring.py
hamiltonian.py
knapsack.py
lcm non recursive.py
lcm recursive.py
map coloring.py
max and min.py
max non recursive.py
max recusive.py
mergesort.py
MST.py
multiplication non recursive.py
multiplication recursive.py
n-queens.py
optimal BST.py
palindrome non recursive.py
palindrome recursive.py
prime or not non recursive.py
prime or not recursive.py
def addEdge(adj, v, w):
adj[v].append(w)
adj[w].append(v)
return adj
def greedyColoring(adj, V):
result = [-1] * V
result[0] = 0;
available = [False] * V
for u in range(1, V):
for i in adj[u]:
if (result[i] != -1):
available[result[i]] = True
cr = 0
while cr < V:
if (available[cr] == False):
break
cr += 1
result[u] = cr
for i in adj[u]:
if (result[i] != -1):
available[result[i]] = False
for u in range(V):
print("Vertex", u, " ---> Color", result[u])
if __name__ == '__main__':
if __name__ == '__main__'


copy the program recursive.py
fact non recursive.py
fact recursive.py
fib non recersive.py
fib recursive.py
floyds.py
gcd non recursive.py
gcd recursive.py
graph colouring.py
hamiltonian.py
knapsack.py
lcm non recursive.py
lcm recursive.py
map coloring.py
max and min.py
max non recursive.py
max recusive.py
mergesort.py
MST.py
multiplication non recursive.py
multiplication recursive.py
n-queens.py
optimal BST.py
palindrome non recursive.py
palindrome recursive.py
hamiltonian.py
B&B Travelling salesman.py
print("Vertex", u, " ---> Color", result[u])
if __name__ == '__main__':
g1 = [[] for i in range(5)]
g1 = addEdge(g1, 0, 1)
g1 = addEdge(g1, 0, 2)
g1 = addEdge(g1, 1, 2)
g1 = addEdge(g1, 1, 3)
g1 = addEdge(g1, 2, 3)
g1 = addEdge(g1, 3, 4)
print("Coloring of graph 1 ")
greedyColoring(g1, 5)
g2 = [[] for i in range(5)]
g2 = addEdge(g2, 0, 1)
g2 = addEdge(g2, 0, 2)
g2 = addEdge(g2, 1, 2)
g2 = addEdge(g2, 1, 4)
g2 = addEdge(g2, 2, 4)
g2 = addEdge(g2, 4, 3)
print("\nColoring of graph 2")
greedyColoring(g2, 5)


hamiltonian.py
B&B Travelling salesman.py
sum of subsets.py
tsp dynamic programmind.py
graph colouring.py
copy the program recursive.py
fact non recursive.py
fact recursive.py
fib non recersive.py
fib recursive.py
floyds.py
gcd non recursive.py
gcd recursive.py
graph colouring.py
hamiltonian.py
knapsack.py
lcm non recursive.py
lcm recursive.py
map coloring.py
max and min.py
max non recursive.py
g2 = addEdge(g2, 1, 2)
g2 = addEdge(g2, 1, 4)
g2 = addEdge(g2, 2, 4)
g2 = addEdge(g2, 4, 3)
print("\nColoring of graph 2")
greedyColoring(g2, 5)
graph colouring
C:\Users\kadiv\PycharmProjects\pythonProject2\venv\Scripts\python.exe "C:/Users/kadiv/AppData/Roaming/JetBrains/PyCharmCE2022.1/scratches/grap
Coloring of graph 1
Vertex 0 ---> Color 0
Vertex 1 ---> Color 1
Vertex 2 ---> Color 2
Vertex 3 ---> Color 0
Vertex 4 ---> Color 1