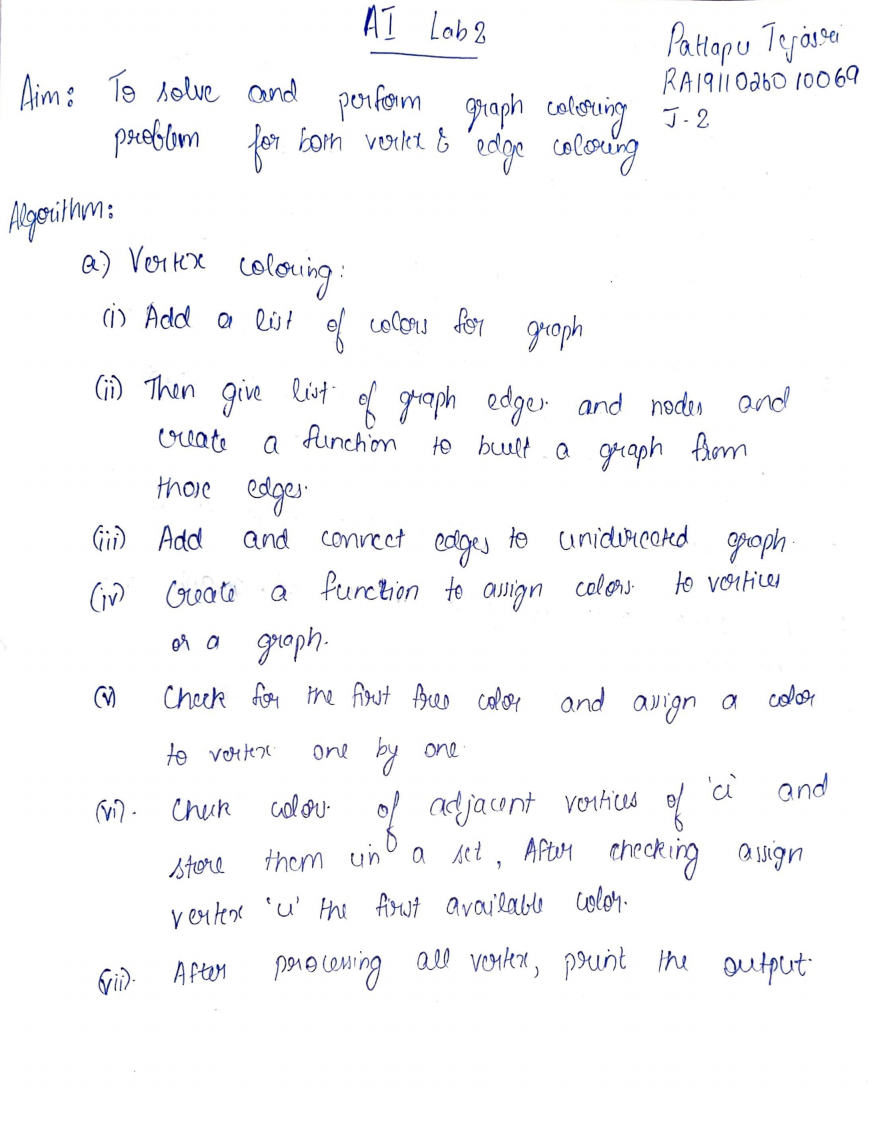
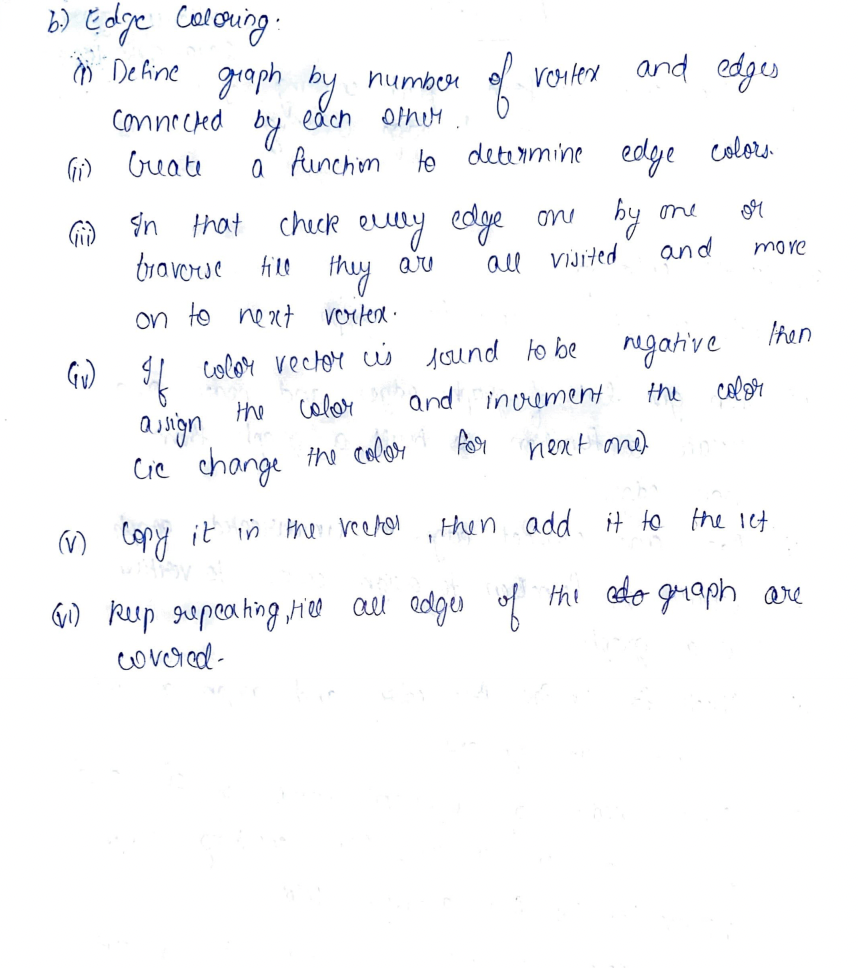
AI LAB EXPERIMENT-2

PATTAPU TEJA SRI

RA1911026010069

J2





Vertex coloring

class Graph:

def \_\_init\_\_(self, edges, n):

self.adjList = [[] for \_ in range(n)]

for (src, dest) in edges:

self.adjList[src].append(dest)

self.adjList[dest].append(src)

def colorGraph(graph, n):

result = {}

for u in range(n):

assigned = set([result.get(i) for i in graph.adjList[u] if i in result])

color = 1

for c in assigned:

if color != c:

break

color = color + 1

result[u] = color

for v in range(n):

print(f'Color assigned to vertex {v} is {colors[result[v]]}')

if \_\_name\_\_ == '\_\_main\_\_':

colors = ['', 'RED', 'YELLOW', 'BLUE', 'GREEN', 'PURPLE', 'PINK',

'BLACK', 'BROWN', 'WHITE', 'ORANGE', 'VOILET']

edges = [(0, 1), (0, 2), (1, 5), (2, 5), (2, 4), (2, 3), (0, 3), (1, 4)]

n = 6

graph = Graph(edges, n)

colorGraph(graph, n)

Edge coloring

from queue import Queue

def colorEdges(ptr, gra, edgeColors, isVisited):

q=Queue()

c = 0

colored=set()

if (isVisited[ptr]):

return

isVisited[ptr] = True

for i in range(len(gra[ptr])) :

if (edgeColors[gra[ptr][i][1]] != -1):

colored.add(edgeColors[gra[ptr][i][1]])

for i in range(len(gra[ptr])) :

if not isVisited[gra[ptr][i][0]]:

q.put(gra[ptr][i][0])

if (edgeColors[gra[ptr][i][1]] == -1) :

while c in colored:

c+=1

edgeColors[gra[ptr][i][1]] = c

colored.add(c)

c+=1

while not q.empty() :

temp = q.get()

colorEdges(temp, gra, edgeColors, isVisited)

return

if \_\_name\_\_=='\_\_main\_\_':

empty=set()

gra=[]

edgeColors=[]

isVisited=[False]\*100000

ver = 4

edge = 4

gra=[[] for \_ in range(ver)]

edgeColors=[-1]\*edge

gra[0].append((1, 0))

gra[1].append((0, 0))

gra[1].append((2, 0))

gra[2].append((1, 1))

gra[2].append((1, 2))

gra[3].append((2, 2))

gra[0].append((0, 3))

gra[3].append((0, 3))

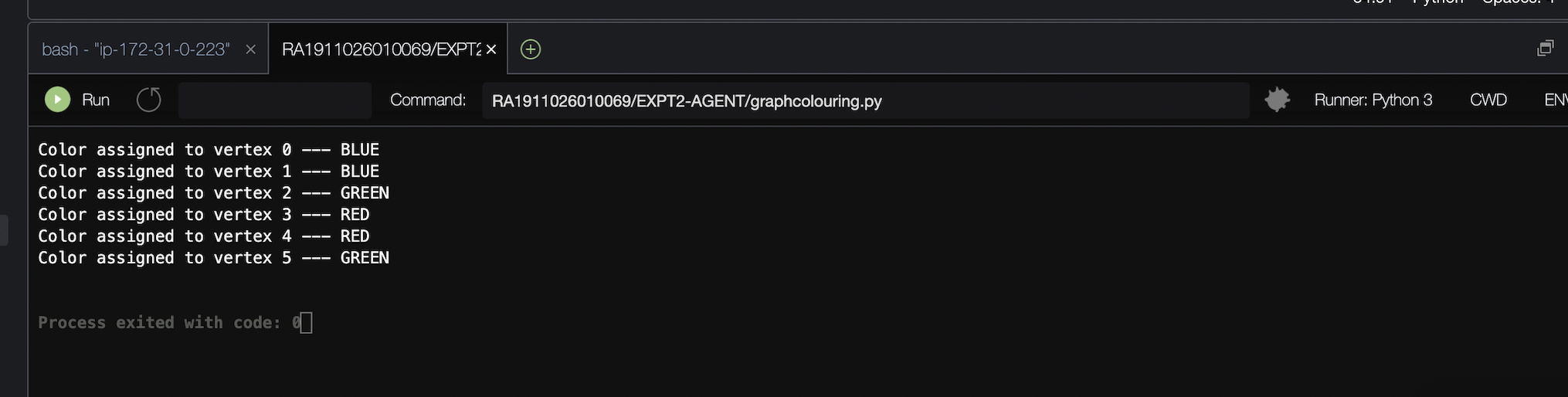
colorEdges(0, gra, edgeColors, isVisited)

for i in range(edge):

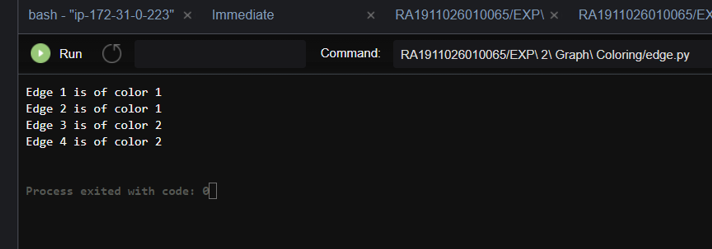
print("Edge {} is of color {}".format(i + 1,edgeColors[i] + 1))

**Output:**

**Vertex colouring:**



**Edge coloring:**



**Result:**

Hence Graph Coloring problem for both vertex coloring and Edge coloring were studied and solved.