

Program 11

Map colouring to implement CSP

AIM :

To implement a prolog program for map colouring to implement CSP.

PROGRAM :

```
def is_safe(graph, color, v, assigned):

    for neighbor in range(len(graph[v])):
        if graph[v][neighbor] == 1 and assigned[neighbor] == color:
            return False
    return True

def solve_map_coloring(graph, colors, m, assigned, v):

    if v == len(graph):
        return True
    for c in range(m):
        if is_safe(graph, colors[c], v, assigned):
            assigned[v] = colors[c]

            if solve_map_coloring(graph, colors, m, assigned, v + 1):
                return True
            assigned[v] = 0

    return False

graph = [
    [0, 1, 1, 1],
    [1, 0, 1, 0],
    [1, 1, 0, 1],
    [1, 0, 1, 0]
]

colors = ["Red", "Green", "Blue", "Yellow"]

m = len(colors)

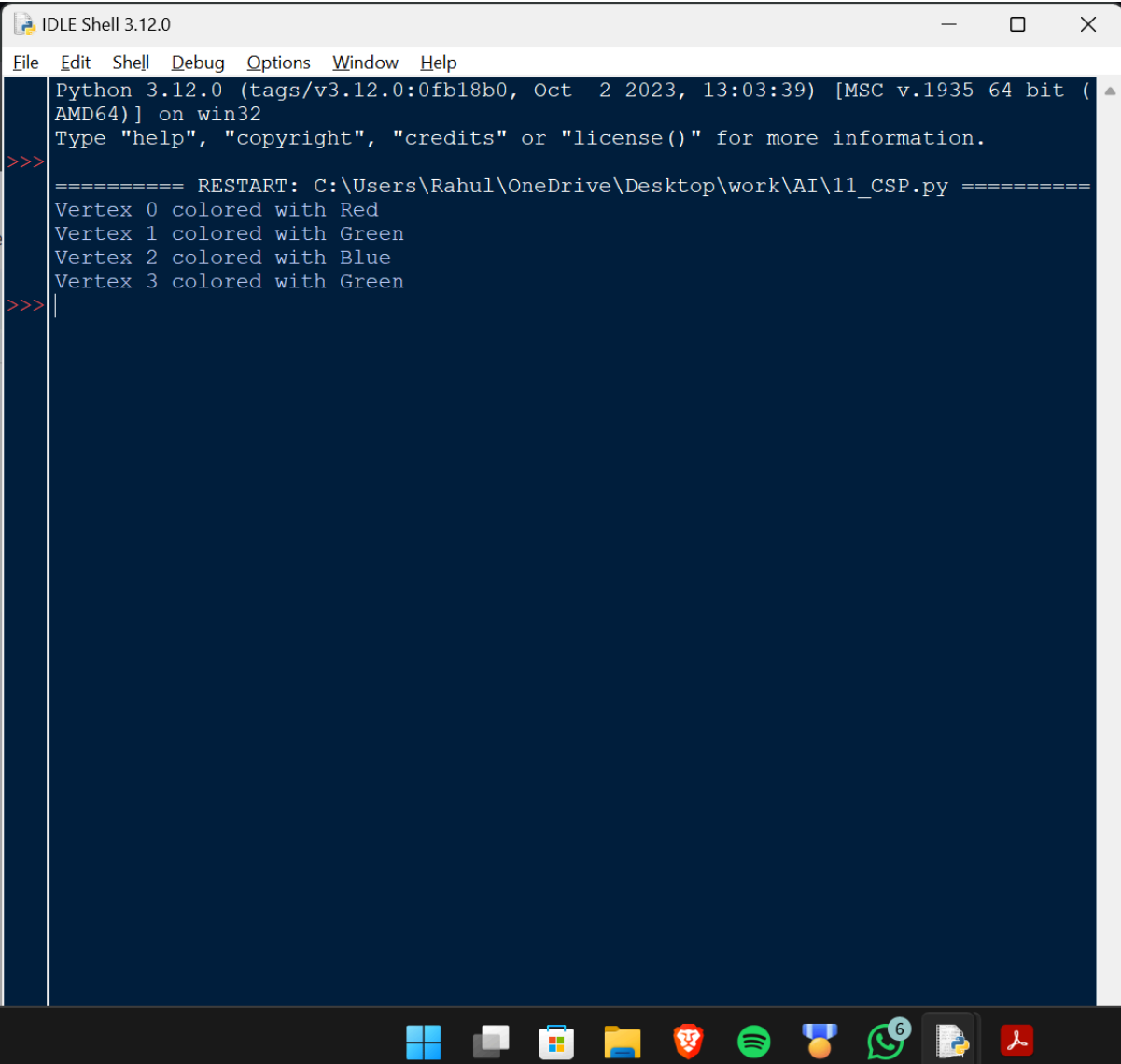
assigned = [0] * len(graph)

if solve_map_coloring(graph, colors, m, assigned, 0):

    for i in range(len(assigned)):
        print(f"Vertex {i} colored with {assigned[i]}")
    else:
```

```
print("No solution found")
```

OUTPUT:



```
IDLE Shell 3.12.0
File Edit Shell Debug Options Window Help
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Rahul\OneDrive\Desktop\work\AI\11_CSP.py =====
Vertex 0 colored with Red
Vertex 1 colored with Green
Vertex 2 colored with Blue
Vertex 3 colored with Green
>>>
```

RESULT:

The Program has successfully been executed.