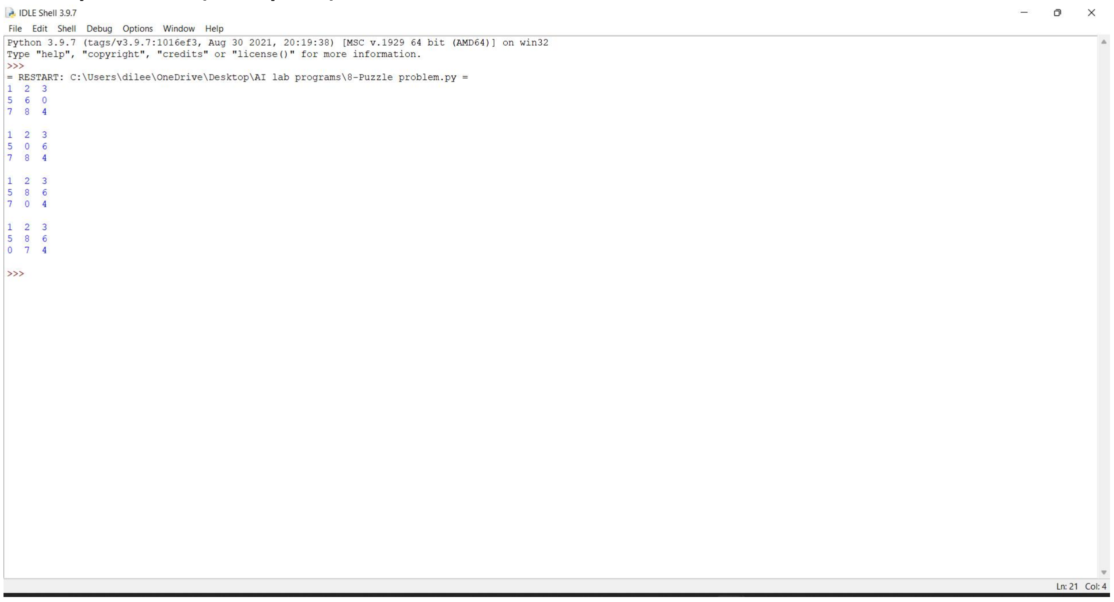
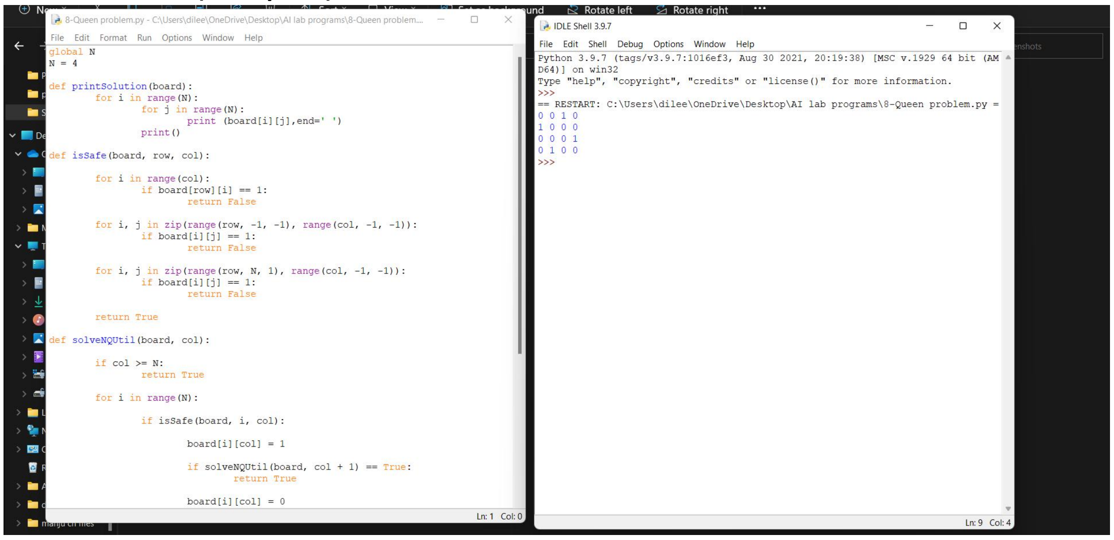
OUTPUTS

1. 8 puzzle (output):



```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\8-Puzzle problem.py =
1  2  3
5  6  0
7  8  4

1  2  3
5  0  6
7  8  4

1  2  3
5  8  6
7  0  4

1  2  3
5  8  6
0  7  4

>>>
```

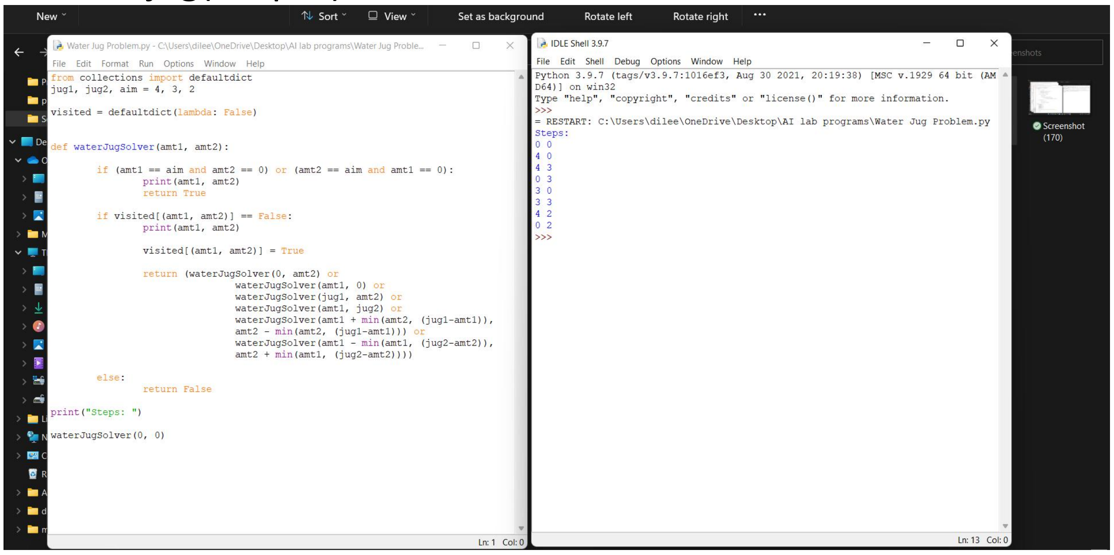
2.8 Queen (output):



```
global N
N = 4

def printSolution(board):
        for i in range(N):
                for j in range(N):
                        print (board[i][j],end=' ')
                print()

def isSafe(board, row, col):

        for i in range(col):
                if board[row][i] == 1:
                        return False

        for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
                if board[i][j] == 1:
                        return False

        for i, j in zip(range(row, N, 1), range(col, -1, -1)):
                if board[i][j] == 1:
                        return False

        return True

def solveNQUtil(board, col):

        if col >= N:
                return True

        for i in range(N):

                if isSafe(board, i, col):

                        board[i][col] = 1

                        if solveNQUtil(board, col + 1) == True:
                                return True

                        board[i][col] = 0
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\8-Queen problem.py =
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
>>>
```

3.water jug(output):



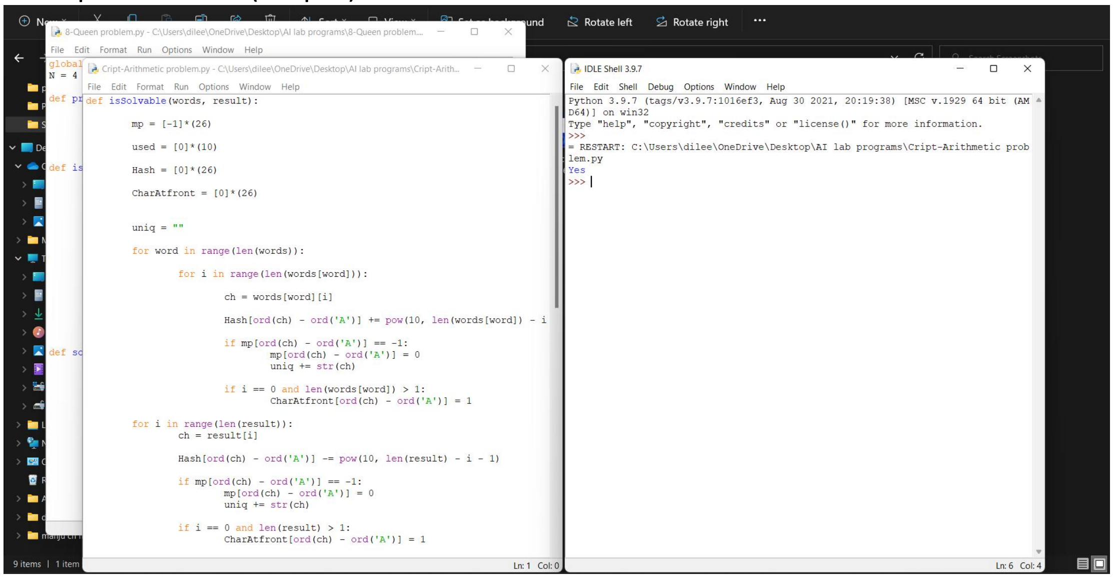
Water Jug Problem.py - C:\Users\dilee\OneDrive\Desktop\AI lab programs\Water Jug Proble...

```python
from collections import defaultdict
jug1, jug2, aim = 4, 3, 2

visited = defaultdict(lambda: False)


def waterJugSolver(amt1, amt2):

        if (amt1 == aim and amt2 == 0) or (amt2 == aim and amt1 == 0):
                print(amt1, amt2)
                return True

        if visited[(amt1, amt2)] == False:
                print(amt1, amt2)

                visited[(amt1, amt2)] = True

                return (waterJugSolver(0, amt2) or
                                waterJugSolver(amt1, 0) or
                                waterJugSolver(jug1, amt2) or
                                waterJugSolver(amt1, jug2) or
                                waterJugSolver(amt1 + min(amt2, (jug1-amt1)),
                                amt2 - min(amt2, (jug1-amt1))) or
                                waterJugSolver(amt1 - min(amt1, (jug2-amt2)),
                                amt2 + min(amt1, (jug2-amt2))))

        else:
                return False

print("Steps: ")

waterJugSolver(0, 0)
```

IDLE Shell 3.9.7

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\Water Jug Problem.py
Steps:
0 0
4 0
4 3
0 3
3 0
3 3
4 2
0 2
>>>
```

4. cript-arithmetic(output):



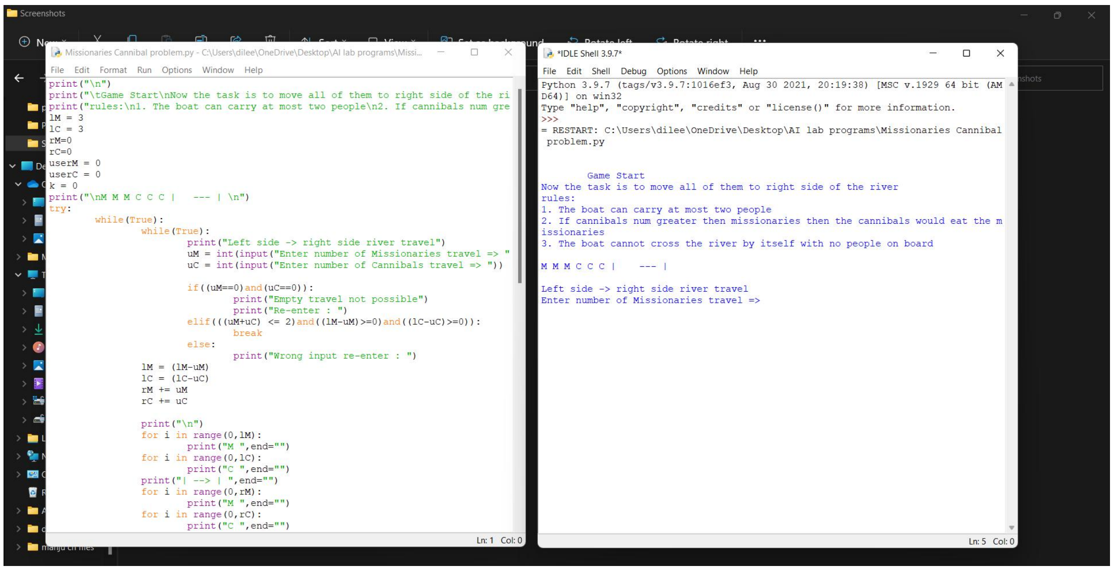
Cript-Arithmetic problem.py - C:\Users\dilee\OneDrive\Desktop\AI lab programs\Cript-Arith...

```python
def isSolvable(words, result):

        mp = [-1]*(26)

        used = [0]*(10)

        Hash = [0]*(26)

        CharAtfront = [0]*(26)


        uniq = ""

        for word in range(len(words)):

                for i in range(len(words[word])):

                        ch = words[word][i]

                        Hash[ord(ch) - ord('A')] += pow(10, len(words[word]) - i

                        if mp[ord(ch) - ord('A')] == -1:
                                mp[ord(ch) - ord('A')] = 0
                                uniq += str(ch)

                        if i == 0 and len(words[word]) > 1:
                                CharAtfront[ord(ch) - ord('A')] = 1

        for i in range(len(result)):
                ch = result[i]

                Hash[ord(ch) - ord('A')] -= pow(10, len(result) - i - 1)

                if mp[ord(ch) - ord('A')] == -1:
                        mp[ord(ch) - ord('A')] = 0
                        uniq += str(ch)

                if i == 0 and len(result) > 1:
                        CharAtfront[ord(ch) - ord('A')] = 1
```

IDLE Shell 3.9.7

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\Cript-Arithmetic prob
lem.py
Yes
>>>
```

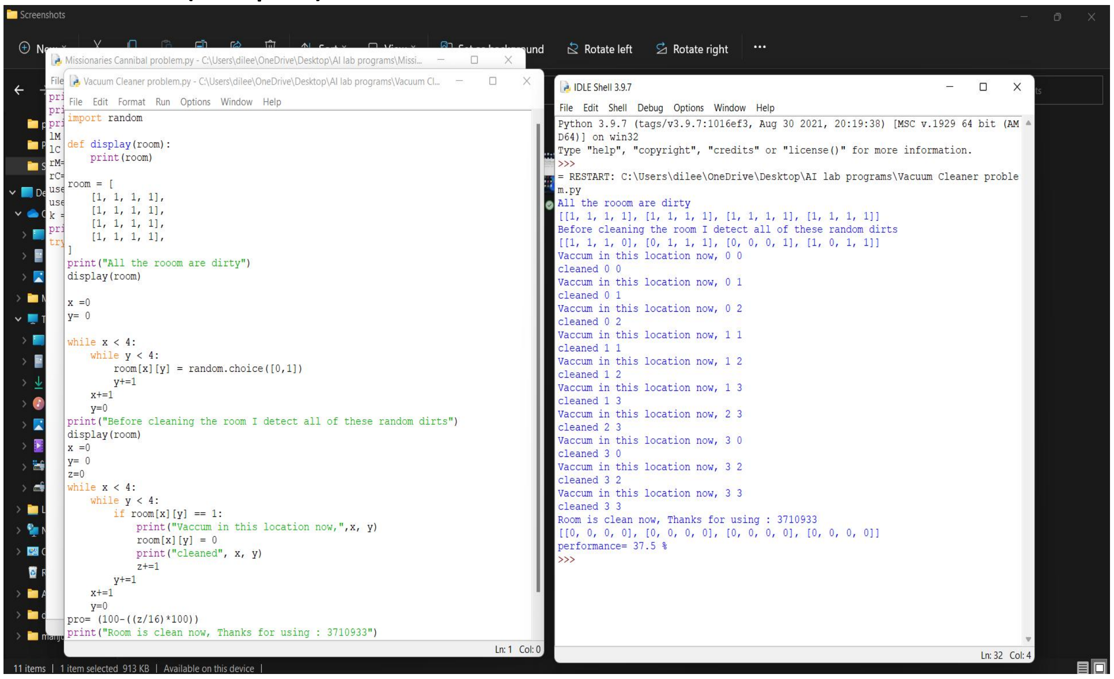
5. Missionaries(output):



```
print("\n")
print("\tGame Start\nNow the task is to move all of them to right side of the ri
print("rules:\n1. The boat can carry at most two people\n2. If cannibals num gre
lM = 3
lC = 3
rM=0
rC=0
userM = 0
userC = 0
k = 0
print("\nM M M C C C |   --- | \n")
try:
        while(True):
                while(True):
                        print("Left side -> right side river travel")
                        uM = int(input("Enter number of Missionaries travel => "
                        uC = int(input("Enter number of Cannibals travel => "))

                        if((uM==0)and(uC==0)):
                                print("Empty travel not possible")
                                print("Re-enter : ")
                        elif(((uM+uC) <= 2)and((lM-uM)>=0)and((lC-uC)>=0)):
                                break
                        else:
                                print("Wrong input re-enter : ")
                lM = (lM-uM)
                lC = (lC-uC)
                rM += uM
                rC += uC

                print("\n")
                for i in range(0,lM):
                        print("M ",end="")
                for i in range(0,lC):
                        print("C ",end="")
                print("| --> | ",end="")
                for i in range(0,rM):
                        print("M ",end="")
                for i in range(0,rC):
                        print("C ",end="")
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\Missionaries Cannibal
 problem.py


        Game Start
Now the task is to move all of them to right side of the river
rules:
1. The boat can carry at most two people
2. If cannibals num greater then missionaries then the cannibals would eat the m
issionaries
3. The boat cannot cross the river by itself with no people on board

M M M C C C |    --- |

Left side -> right side river travel
Enter number of Missionaries travel =>
```

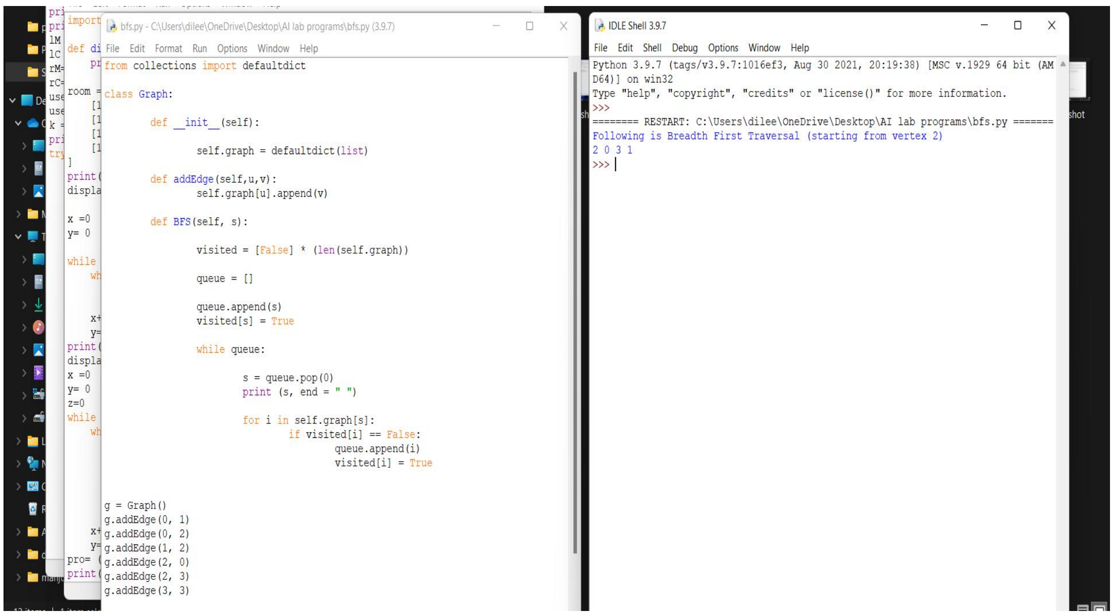
6. Vaccum(output):



```
import random

def display(room):
    print(room)

room = [
    [1, 1, 1, 1],
    [1, 1, 1, 1],
    [1, 1, 1, 1],
    [1, 1, 1, 1],
]
print("All the rooom are dirty")
display(room)

x =0
y= 0

while x < 4:
    while y < 4:
        room[x][y] = random.choice([0,1])
        y+=1
    x+=1
    y=0
print("Before cleaning the room I detect all of these random dirts")
display(room)
x =0
y= 0
z=0
while x < 4:
    while y < 4:
        if room[x][y] == 1:
            print("Vaccum in this location now,",x, y)
            room[x][y] = 0
            print("cleaned", x, y)
            z+=1
        y+=1
    x+=1
    y=0
pro= (100-((z/16)*100))
print("Room is clean now, Thanks for using : 3710933")
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\Vacuum Cleaner proble
m.py
All the rooom are dirty
[[1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1], [1, 1, 1, 1]]
Before cleaning the room I detect all of these random dirts
[[1, 1, 1, 0], [0, 1, 1, 1], [0, 0, 0, 1], [1, 0, 1, 1]]
Vaccum in this location now, 0 0
cleaned 0 0
Vaccum in this location now, 0 1
cleaned 0 1
Vaccum in this location now, 0 2
cleaned 0 2
Vaccum in this location now, 1 1
cleaned 1 1
Vaccum in this location now, 1 2
cleaned 1 2
Vaccum in this location now, 1 3
cleaned 1 3
Vaccum in this location now, 2 3
cleaned 2 3
Vaccum in this location now, 3 0
cleaned 3 0
Vaccum in this location now, 3 2
cleaned 3 2
Vaccum in this location now, 3 3
cleaned 3 3
Room is clean now, Thanks for using : 3710933
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
performance= 37.5 %
>>>
```

7. Bfs(output):



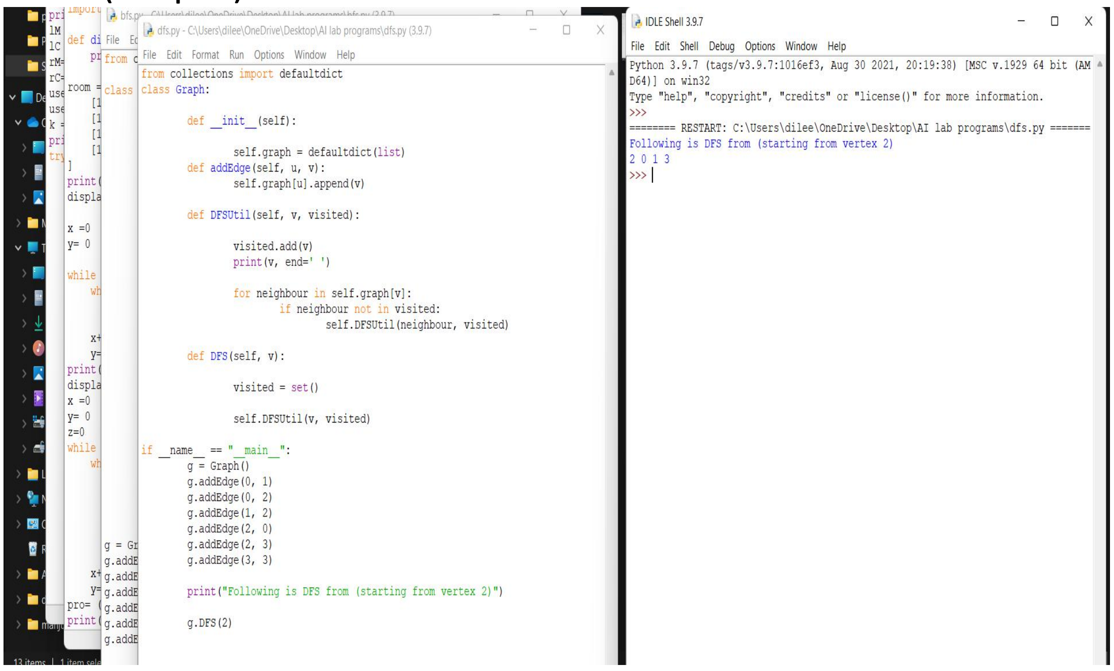
bfs.py - C:\Users\dilee\OneDrive\Desktop\AI lab programs\bfs.py (3.9.7)

```
from collections import defaultdict

class Graph:

        def __init__(self):

                self.graph = defaultdict(list)

        def addEdge(self,u,v):
                self.graph[u].append(v)

        def BFS(self, s):

                visited = [False] * (len(self.graph))

                queue = []

                queue.append(s)
                visited[s] = True

                while queue:

                        s = queue.pop(0)
                        print (s, end = " ")

                        for i in self.graph[s]:
                                if visited[i] == False:
                                        queue.append(i)
                                        visited[i] = True


g = Graph()
g.addEdge(0, 1)
g.addEdge(0, 2)
g.addEdge(1, 2)
g.addEdge(2, 0)
g.addEdge(2, 3)
g.addEdge(3, 3)
```

IDLE Shell 3.9.7

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\bfs.py =======
Following is Breadth First Traversal (starting from vertex 2)
2 0 3 1
>>>
```

8. Dfs(output):



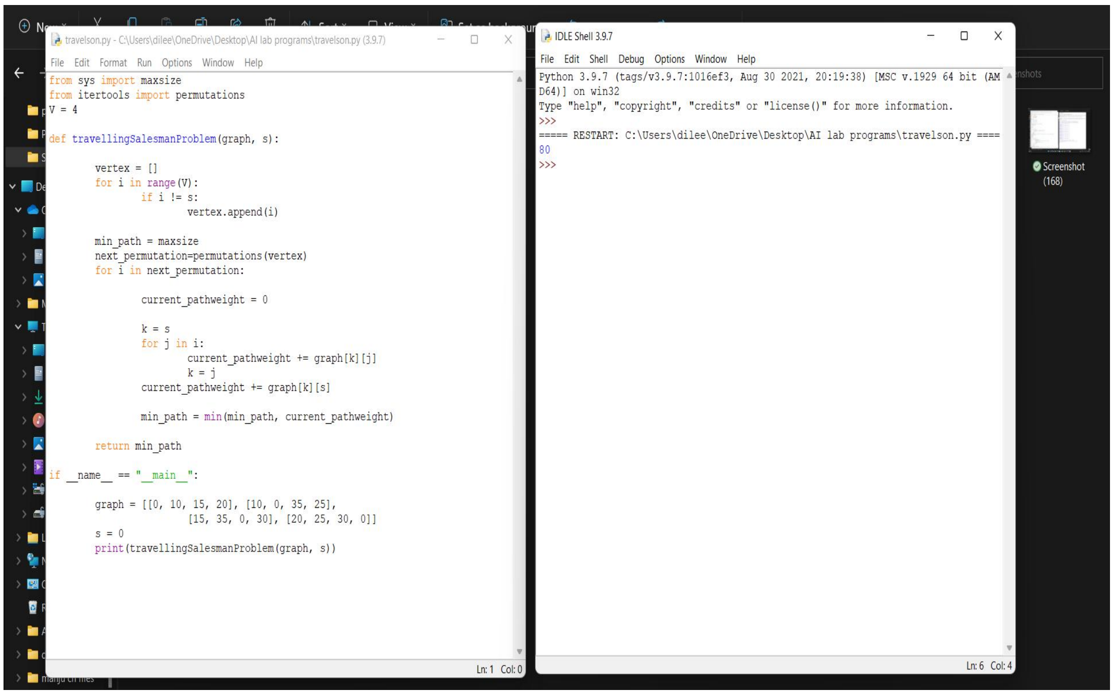
dfs.py - C:\Users\dilee\OneDrive\Desktop\AI lab programs\dfs.py (3.9.7)

```
from collections import defaultdict
class Graph:

        def __init__(self):

                self.graph = defaultdict(list)
        def addEdge(self, u, v):
                self.graph[u].append(v)

        def DFSUtil(self, v, visited):

                visited.add(v)
                print(v, end=' ')

                for neighbour in self.graph[v]:
                        if neighbour not in visited:
                                self.DFSUtil(neighbour, visited)

        def DFS(self, v):

                visited = set()

                self.DFSUtil(v, visited)

if __name__ == "__main__":
        g = Graph()
        g.addEdge(0, 1)
        g.addEdge(0, 2)
        g.addEdge(1, 2)
        g.addEdge(2, 0)
        g.addEdge(2, 3)
        g.addEdge(3, 3)

        print("Following is DFS from (starting from vertex 2)")

        g.DFS(2)
```

IDLE Shell 3.9.7

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\dfs.py =======
Following is DFS from (starting from vertex 2)
2 0 1 3
>>>
```

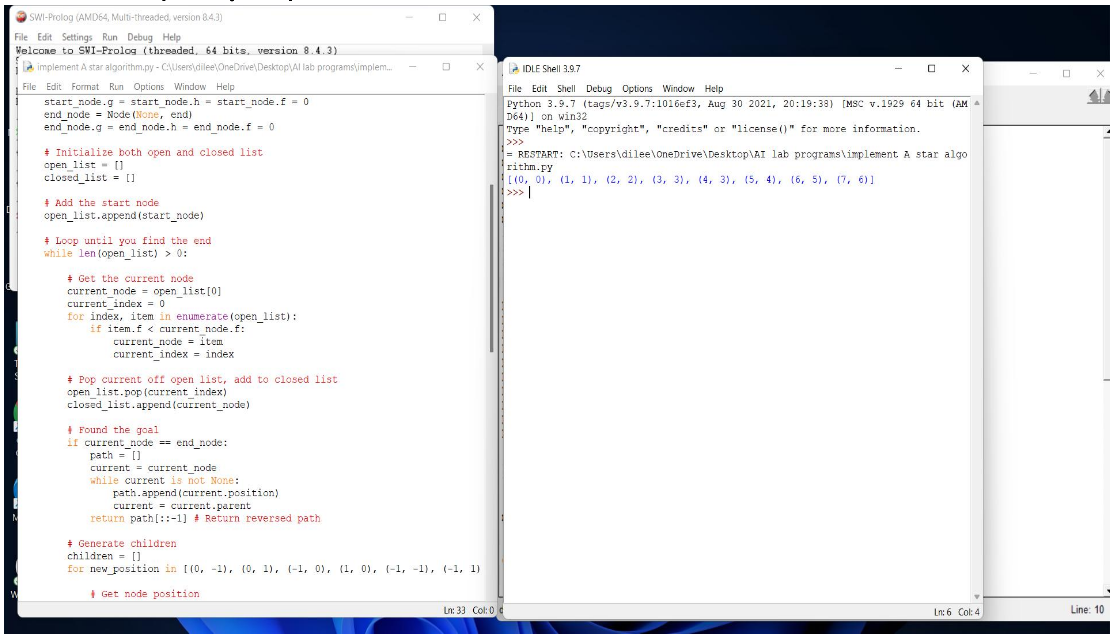
9. travelson(output):



```
from sys import maxsize
from itertools import permutations
V = 4

def travellingSalesmanProblem(graph, s):

    vertex = []
    for i in range(V):
        if i != s:
            vertex.append(i)

    min_path = maxsize
    next_permutation=permutations(vertex)
    for i in next_permutation:

        current_pathweight = 0

        k = s
        for j in i:
            current_pathweight += graph[k][j]
            k = j
        current_pathweight += graph[k][s]

        min_path = min(min_path, current_pathweight)

    return min_path

if __name__ == "__main__":

    graph = [[0, 10, 15, 20], [10, 0, 35, 25],
            [15, 35, 0, 30], [20, 25, 30, 0]]
    s = 0
    print(travellingSalesmanProblem(graph, s))
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\travelson.py ====
80
>>>
```

10. A* (output):

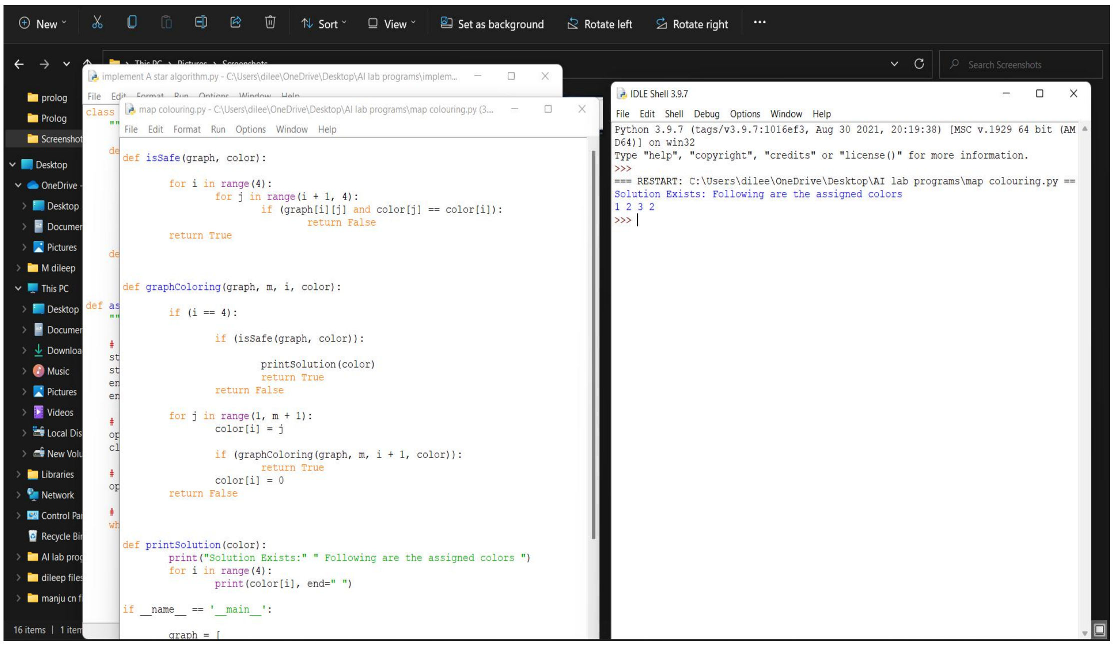


```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
```

```
    start_node.g = start_node.h = start_node.f = 0
    end_node = Node(None, end)
    end_node.g = end_node.h = end_node.f = 0

    # Initialize both open and closed list
    open_list = []
    closed_list = []

    # Add the start node
    open_list.append(start_node)

    # Loop until you find the end
    while len(open_list) > 0:

        # Get the current node
        current_node = open_list[0]
        current_index = 0
        for index, item in enumerate(open_list):
            if item.f < current_node.f:
                current_node = item
                current_index = index

        # Pop current off open list, add to closed list
        open_list.pop(current_index)
        closed_list.append(current_node)

        # Found the goal
        if current_node == end_node:
            path = []
            current = current_node
            while current is not None:
                path.append(current.position)
                current = current.parent
            return path[::-1] # Return reversed path

        # Generate children
        children = []
        for new_position in [(0, -1), (0, 1), (-1, 0), (1, 0), (-1, -1), (-1, 1)

            # Get node position
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\implement A star algo
rithm.py
[(0, 0), (1, 1), (2, 2), (3, 3), (4, 3), (5, 4), (6, 5), (7, 6)]
>>> 
```

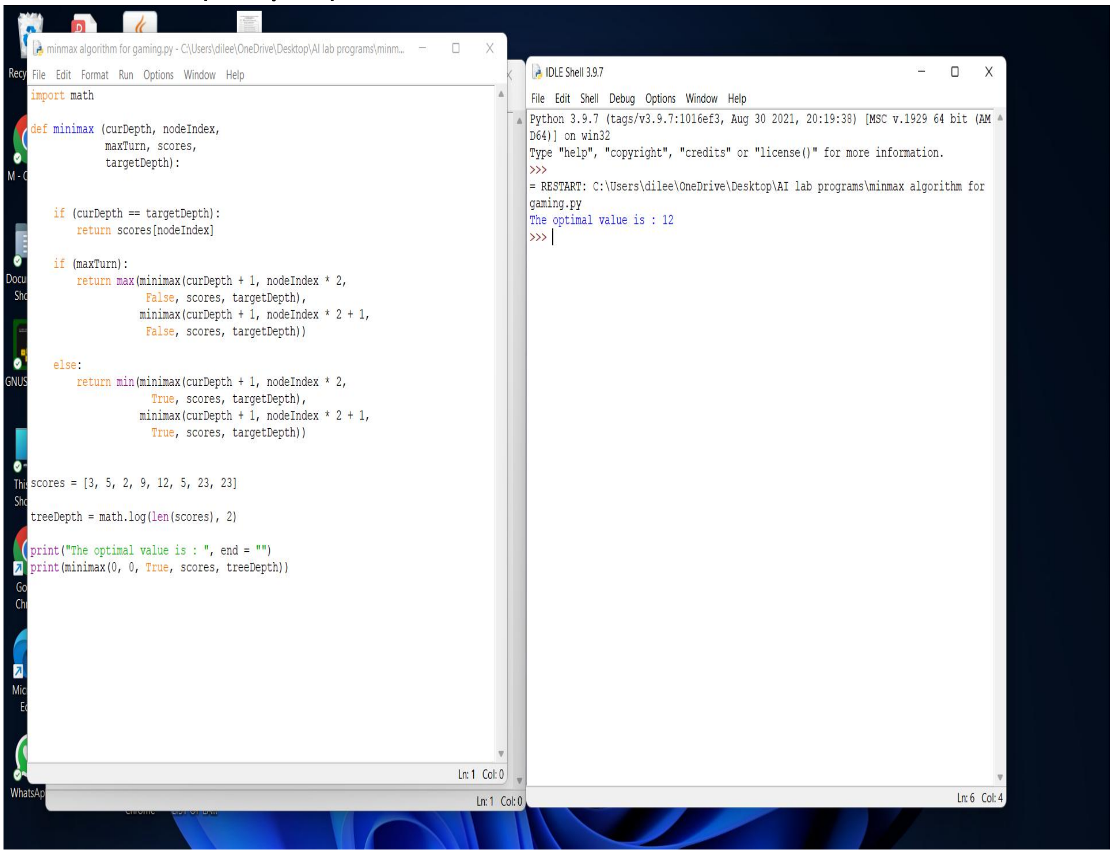
11. map color(output):



```
def isSafe(graph, color):

        for i in range(4):
                for j in range(i + 1, 4):
                        if (graph[i][j] and color[j] == color[i]):
                                return False
        return True



def graphColoring(graph, m, i, color):

        if (i == 4):

                if (isSafe(graph, color)):

                        printSolution(color)
                        return True
                return False

        for j in range(1, m + 1):
                color[i] = j

                if (graphColoring(graph, m, i + 1, color)):
                        return True
                color[i] = 0
        return False



def printSolution(color):
        print("Solution Exists:" " Following are the assigned colors ")
        for i in range(4):
                print(color[i], end=" ")

if __name__ == '__main__':

        graph = [
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\map colouring.py ==
Solution Exists: Following are the assigned colors
1 2 3 2
>>>
```

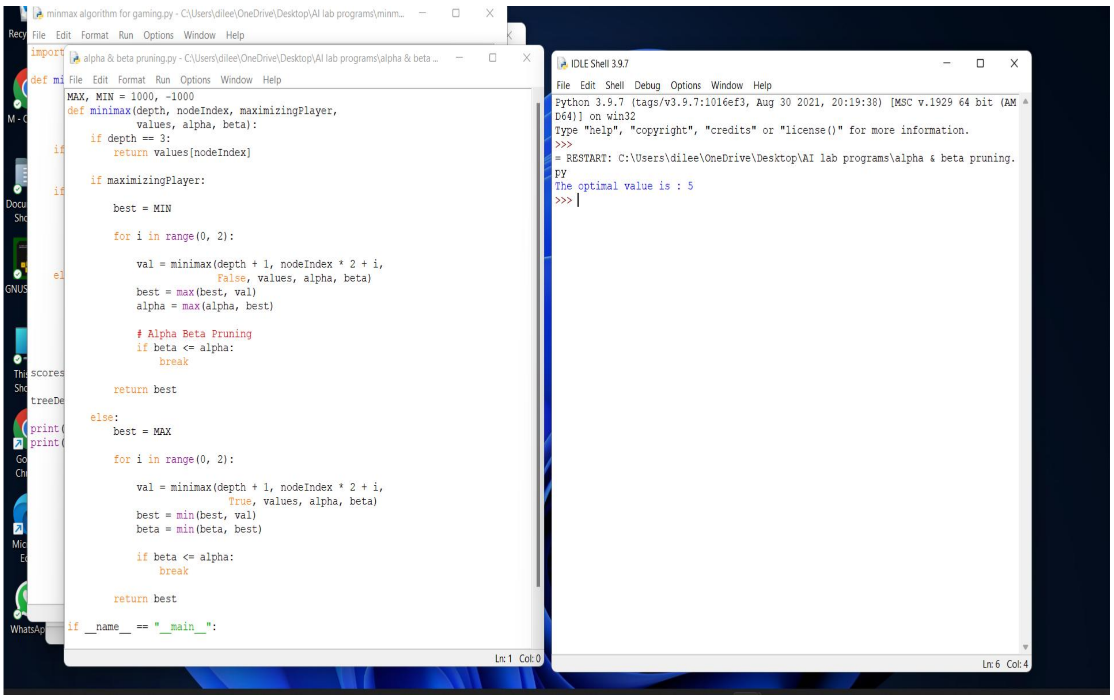
13.minmax(output):



```
import math

def minimax (curDepth, nodeIndex,
             maxTurn, scores,
             targetDepth):


    if (curDepth == targetDepth):
        return scores[nodeIndex]

    if (maxTurn):
        return max(minimax(curDepth + 1, nodeIndex * 2,
                    False, scores, targetDepth),
                   minimax(curDepth + 1, nodeIndex * 2 + 1,
                    False, scores, targetDepth))

    else:
        return min(minimax(curDepth + 1, nodeIndex * 2,
                     True, scores, targetDepth),
                   minimax(curDepth + 1, nodeIndex * 2 + 1,
                     True, scores, targetDepth))


scores = [3, 5, 2, 9, 12, 5, 23, 23]

treeDepth = math.log(len(scores), 2)

print("The optimal value is : ", end = "")
print(minimax(0, 0, True, scores, treeDepth))
```

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\minmax algorithm for
gaming.py
The optimal value is : 12
>>>
```

14. alpha Beta(output):



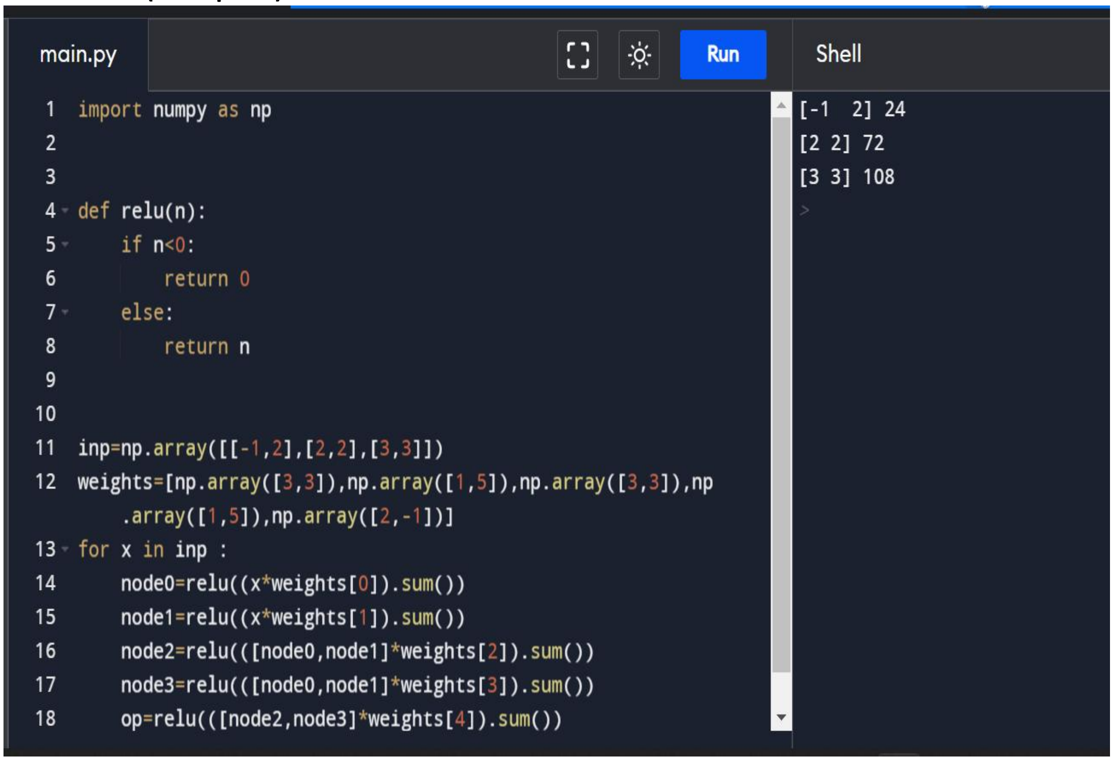
alpha & beta pruning.py - C:\Users\dilee\OneDrive\Desktop\AI lab programs\alpha & beta ...

```python
MAX, MIN = 1000, -1000
def minimax(depth, nodeIndex, maximizingPlayer,
            values, alpha, beta):
    if depth == 3:
        return values[nodeIndex]

    if maximizingPlayer:

        best = MIN

        for i in range(0, 2):

            val = minimax(depth + 1, nodeIndex * 2 + i,
                          False, values, alpha, beta)
            best = max(best, val)
            alpha = max(alpha, best)

            # Alpha Beta Pruning
            if beta <= alpha:
                break

        return best

    else:
        best = MAX

        for i in range(0, 2):

            val = minimax(depth + 1, nodeIndex * 2 + i,
                            True, values, alpha, beta)
            best = min(best, val)
            beta = min(beta, best)

            if beta <= alpha:
                break

        return best

if __name__ == "__main__":
```

IDLE Shell 3.9.7

```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dilee\OneDrive\Desktop\AI lab programs\alpha & beta pruning.
py
The optimal value is : 5
>>> 
```

16.feed forward(output):



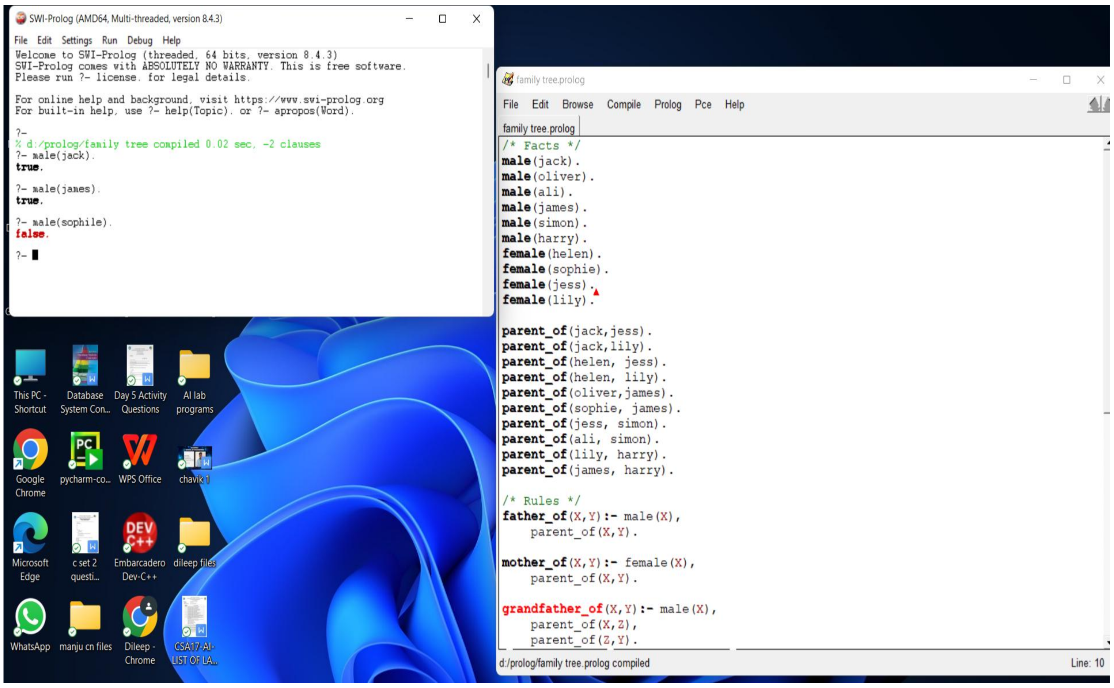
main.py

```python
import numpy as np


def relu(n):
    if n<0:
        return 0
    else:
        return n


inp=np.array([[-1,2],[2,2],[3,3]])
weights=[np.array([3,3]),np.array([1,5]),np.array([3,3]),np
    .array([1,5]),np.array([2,-1])]
for x in inp :
    node0=relu((x*weights[0]).sum())
    node1=relu((x*weights[1]).sum())
    node2=relu(([node0,node1]*weights[2]).sum())
    node3=relu(([node0,node1]*weights[3]).sum())
    op=relu(([node2,node3]*weights[4]).sum())
```

Shell

```
[-1  2] 24
[2 2] 72
[3 3] 108
```

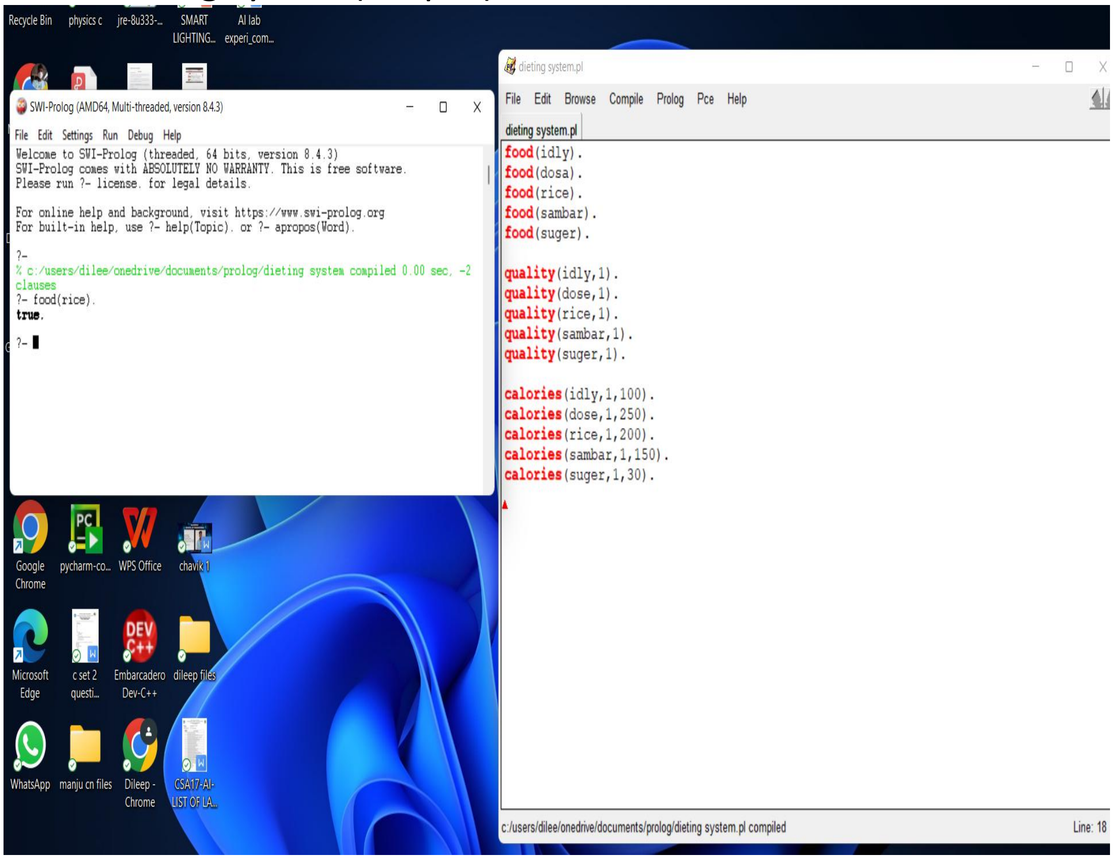
17.family tree(output):



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/prolog/family tree compiled 0.02 sec, -2 clauses
?- male(jack).
true.

?- male(james).
true.

?- male(sophie).
false.

?-
```

```
/* Facts */
male(jack).
male(oliver).
male(ali).
male(james).
male(simon).
male(harry).
female(helen).
female(sophie).
female(jess).
female(lily).

parent_of(jack,jess).
parent_of(jack,lily).
parent_of(helen, jess).
parent_of(helen, lily).
parent_of(oliver,james).
parent_of(sophie, james).
parent_of(jess, simon).
parent_of(ali, simon).
parent_of(lily, harry).
parent_of(james, harry).

/* Rules */
father_of(X,Y):- male(X),
    parent_of(X,Y).

mother_of(X,Y):- female(X),
    parent_of(X,Y).

grandfather_of(X,Y):- male(X),
    parent_of(X,Z),
    parent_of(Z,Y).
```

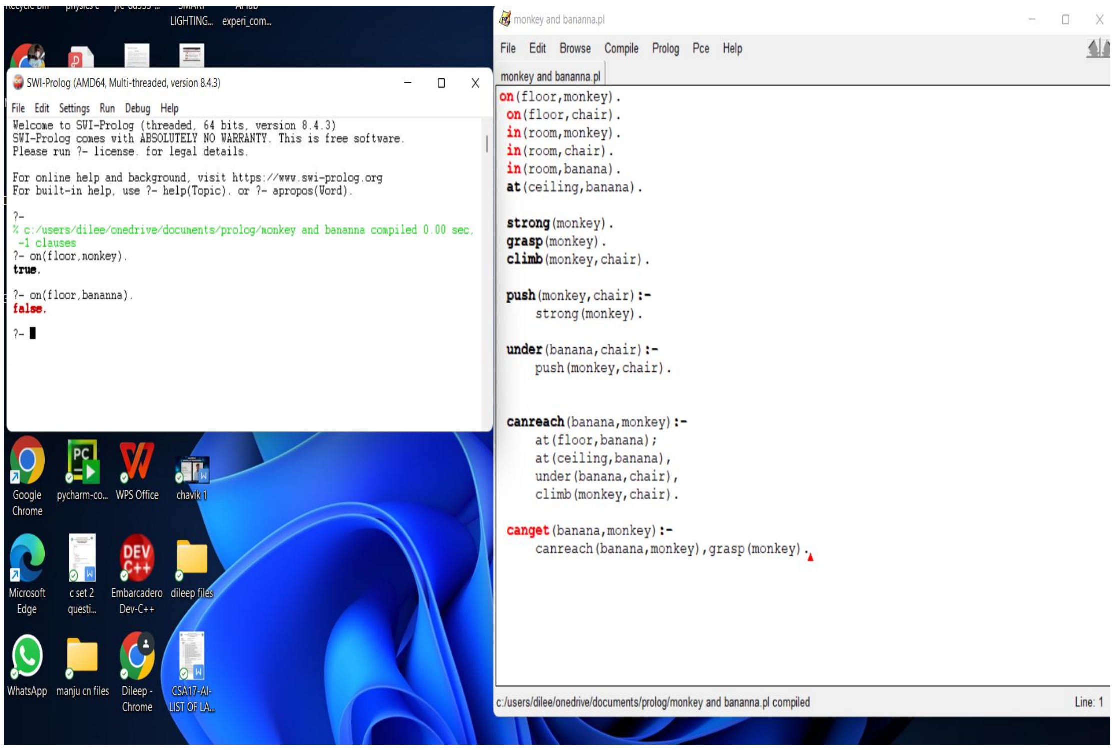
18. dieting system(output):



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/users/dilee/onedrive/documents/prolog/dieting system compiled 0.00 sec, -2
clauses
?- food(rice).
true.

?-
```

```
food(idly).
food(dosa).
food(rice).
food(sambar).
food(suger).

quality(idly,1).
quality(dose,1).
quality(rice,1).
quality(sambar,1).
quality(suger,1).

calories(idly,1,100).
calories(dose,1,250).
calories(rice,1,200).
calories(sambar,1,150).
calories(suger,1,30).
```

19. monkey &Bananna(output):

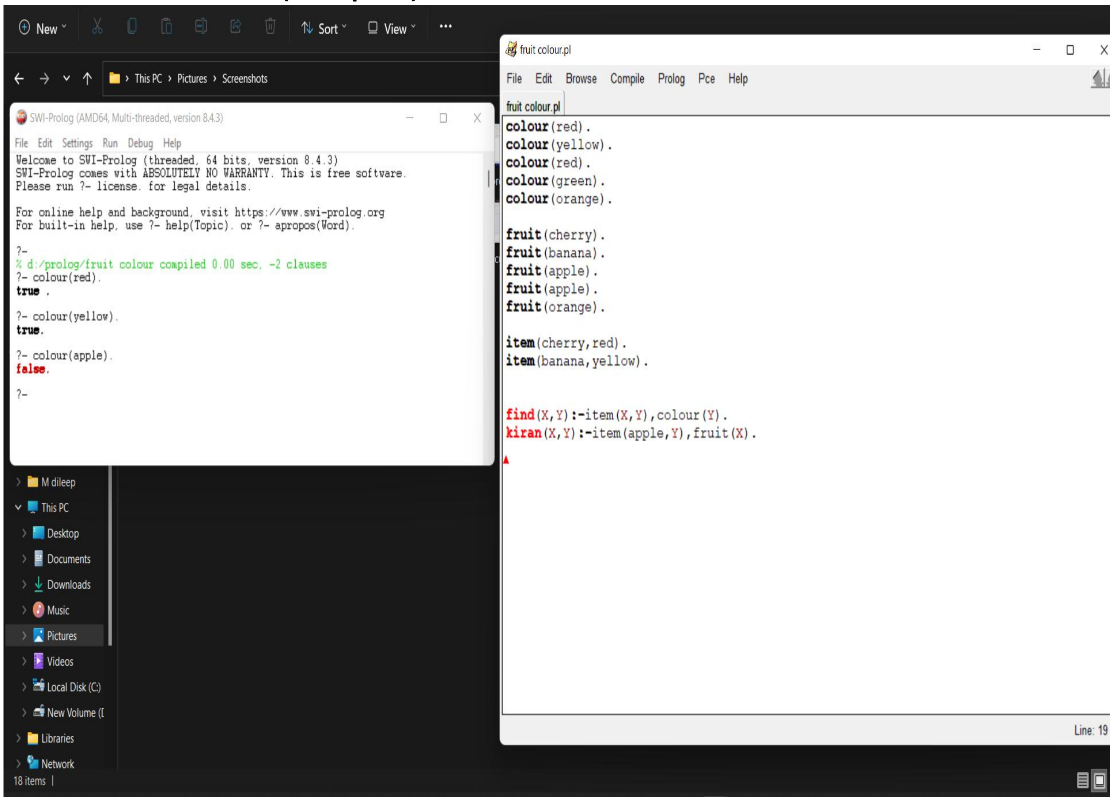


```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/users/dilee/onedrive/documents/prolog/monkey and bananna compiled 0.00 sec,
 -1 clauses
?- on(floor,monkey).
true.

?- on(floor,bananna).
false.

?-
```

```
on(floor,monkey).
on(floor,chair).
in(room,monkey).
in(room,chair).
in(room,banana).
at(ceiling,banana).

strong(monkey).
grasp(monkey).
climb(monkey,chair).

push(monkey,chair):-
    strong(monkey).

under(banana,chair):-
    push(monkey,chair).


canreach(banana,monkey):-
    at(floor,banana);
    at(ceiling,banana),
    under(banana,chair),
    climb(monkey,chair).

canget(banana,monkey):-
    canreach(banana,monkey),grasp(monkey).
```

c:/users/dilee/onedrive/documents/prolog/monkey and bananna.pl compiled   Line: 1

20. Fruit color(output):



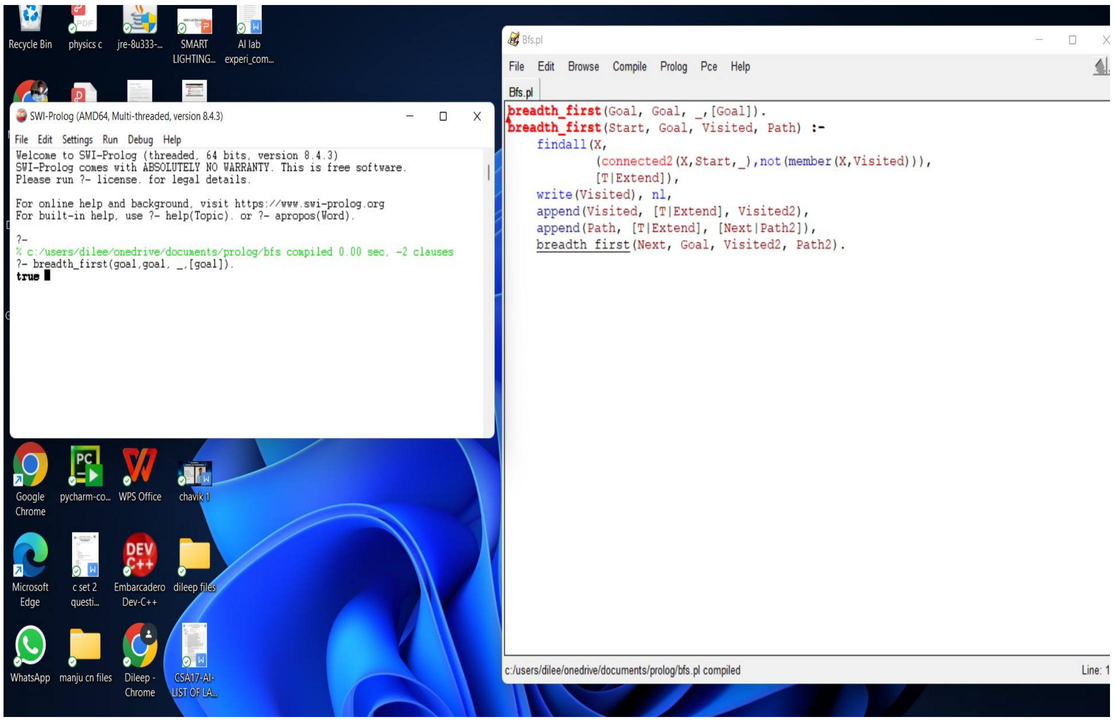
New
Sort
View
This PC › Pictures › Screenshots
SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
fruit colour.pl
File Edit Browse Compile Prolog Pce Help
M dileep
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
New Volume (I
Libraries
Network
18 items

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/prolog/fruit colour compiled 0.00 sec, -2 clauses
?- colour(red).
true .

?- colour(yellow).
true.

?- colour(apple).
false.

?-
```

```
colour(red).
colour(yellow).
colour(red).
colour(green).
colour(orange).

fruit(cherry).
fruit(banana).
fruit(apple).
fruit(apple).
fruit(orange).

item(cherry,red).
item(banana,yellow).


find(X,Y):-item(X,Y),colour(Y).
kiran(X,Y):-item(apple,Y),fruit(X).
```

Line: 19

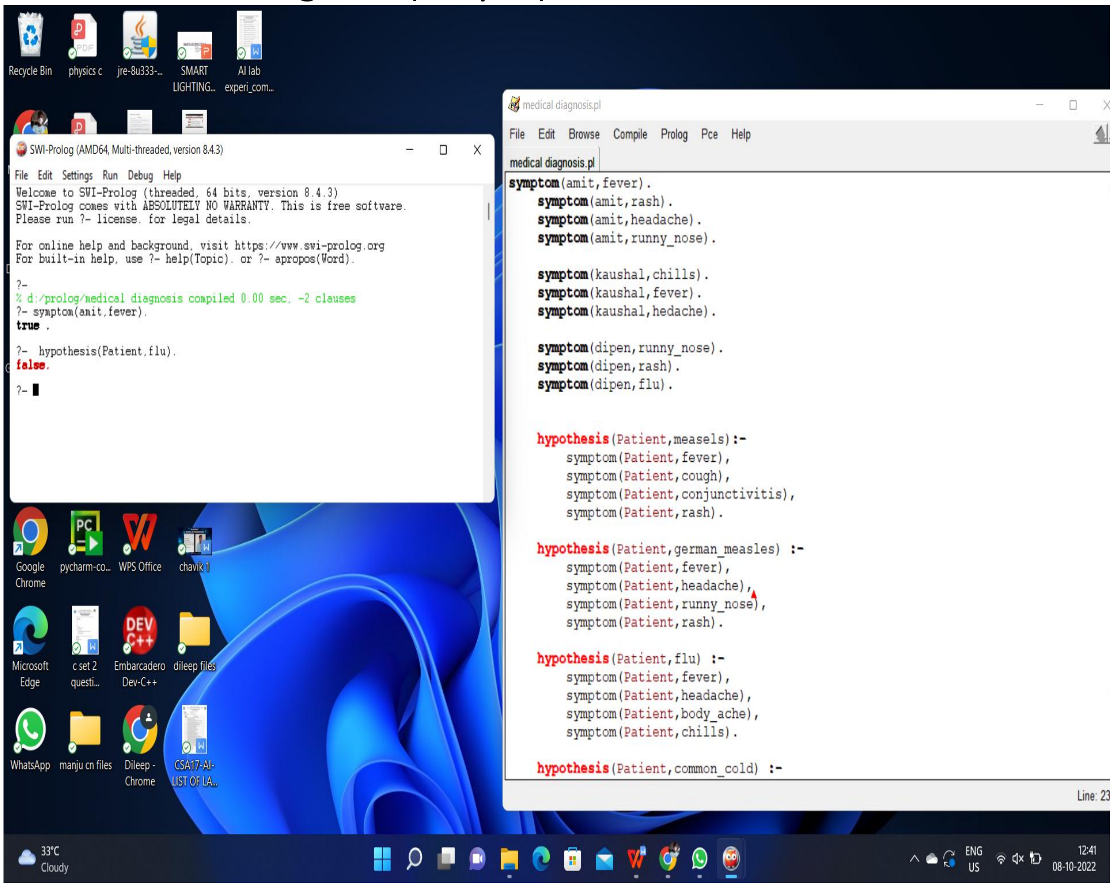
21. Bfs prolog(output):



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/users/dilee/onedrive/documents/prolog/bfs compiled 0.00 sec, -2 clauses
?- breadth_first(goal,goal, _,[goal]).
true
```

```
breadth_first(Goal, Goal, _,[Goal]).
breadth_first(Start, Goal, Visited, Path) :-
    findall(X,
            (connected2(X,Start,_),not(member(X,Visited))),
            [T|Extend]),
    write(Visited), nl,
    append(Visited, [T|Extend], Visited2),
    append(Path, [T|Extend], [Next|Path2]),
    breadth_first(Next, Goal, Visited2, Path2).
```

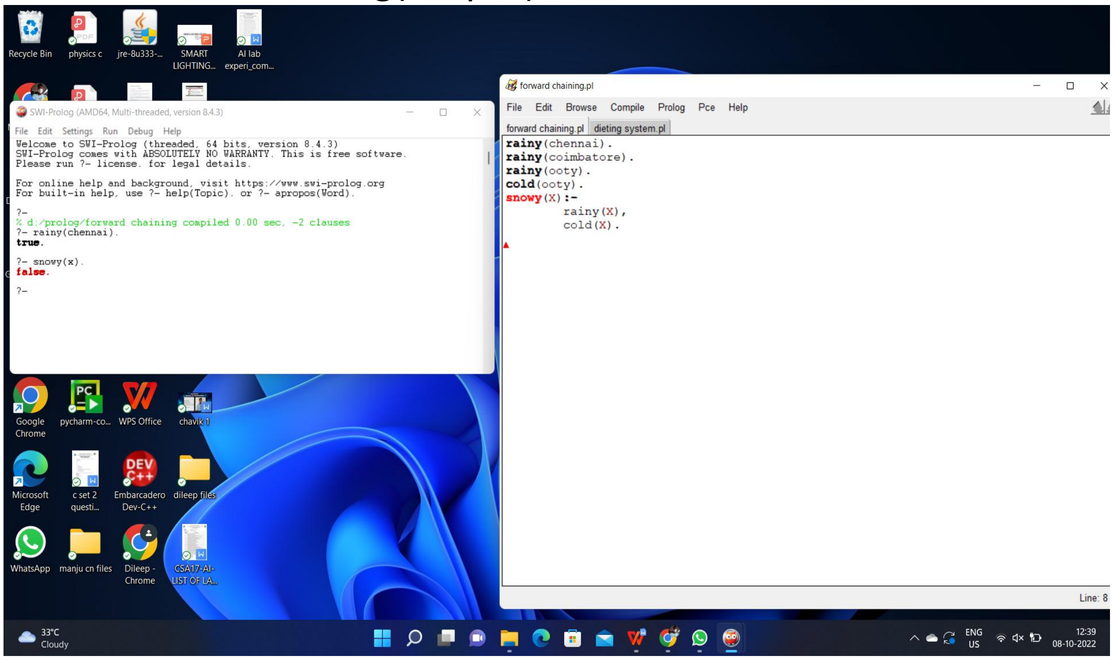
22. medical diagnosis(output):



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/prolog/medical diagnosis compiled 0.00 sec, -2 clauses
?- symptom(amit,fever).
true .

?-  hypothesis(Patient,flu).
false.

?-
```

```
symptom(amit,fever).
    symptom(amit,rash).
    symptom(amit,headache).
    symptom(amit,runny_nose).

    symptom(kaushal,chills).
    symptom(kaushal,fever).
    symptom(kaushal,hedache).

    symptom(dipen,runny_nose).
    symptom(dipen,rash).
    symptom(dipen,flu).


    hypothesis(Patient,measels):-
        symptom(Patient,fever),
        symptom(Patient,cough),
        symptom(Patient,conjunctivitis),
        symptom(Patient,rash).

    hypothesis(Patient,german_measles) :-
        symptom(Patient,fever),
        symptom(Patient,headache),
        symptom(Patient,runny_nose),
        symptom(Patient,rash).

    hypothesis(Patient,flu) :-
        symptom(Patient,fever),
        symptom(Patient,headache),
        symptom(Patient,body_ache),
        symptom(Patient,chills).

    hypothesis(Patient,common_cold) :-
```

23. forward chaining(output):

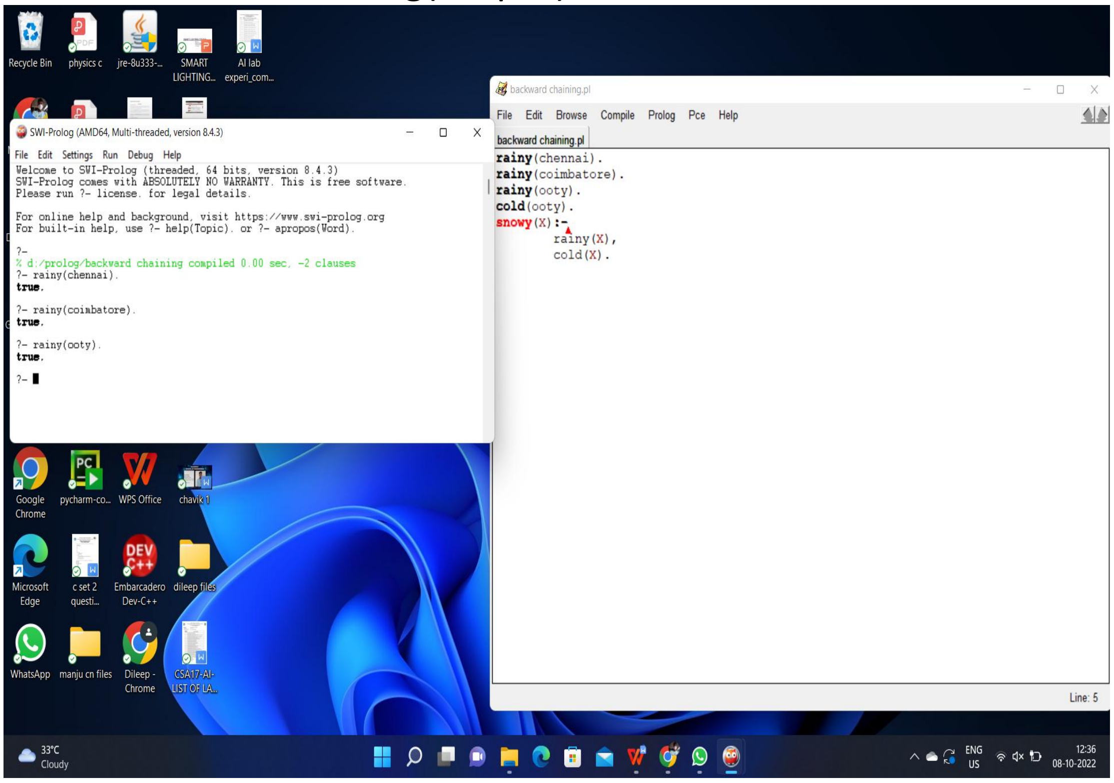


```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/prolog/forward chaining compiled 0.00 sec, -2 clauses
?- rainy(chennai).
true.

?- snowy(x).
false.

?-
```

```
rainy(chennai).
rainy(coimbatore).
rainy(ooty).
cold(ooty).
snowy(X):-
        rainy(X),
        cold(X).
```

SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
forward chaining.pl
File Edit Browse Compile Prolog Pce Help
forward chaining.pl dieting system.pl
Line: 8

24. Backward chaining(output):



```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/prolog/backward chaining compiled 0.00 sec, -2 clauses
?- rainy(chennai).
true.

?- rainy(coimbatore).
true.

?- rainy(ooty).
true.

?-
```

```
rainy(chennai).
rainy(coimbatore).
rainy(ooty).
cold(ooty).
snowy(X):-
        rainy(X),
        cold(X).
```

SWI-Prolog (AMD64, Multi-threaded, version 8.4.3)
File Edit Settings Run Debug Help
backward chaining.pl
File Edit Browse Compile Prolog Pce Help
backward chaining.pl
Line: 5