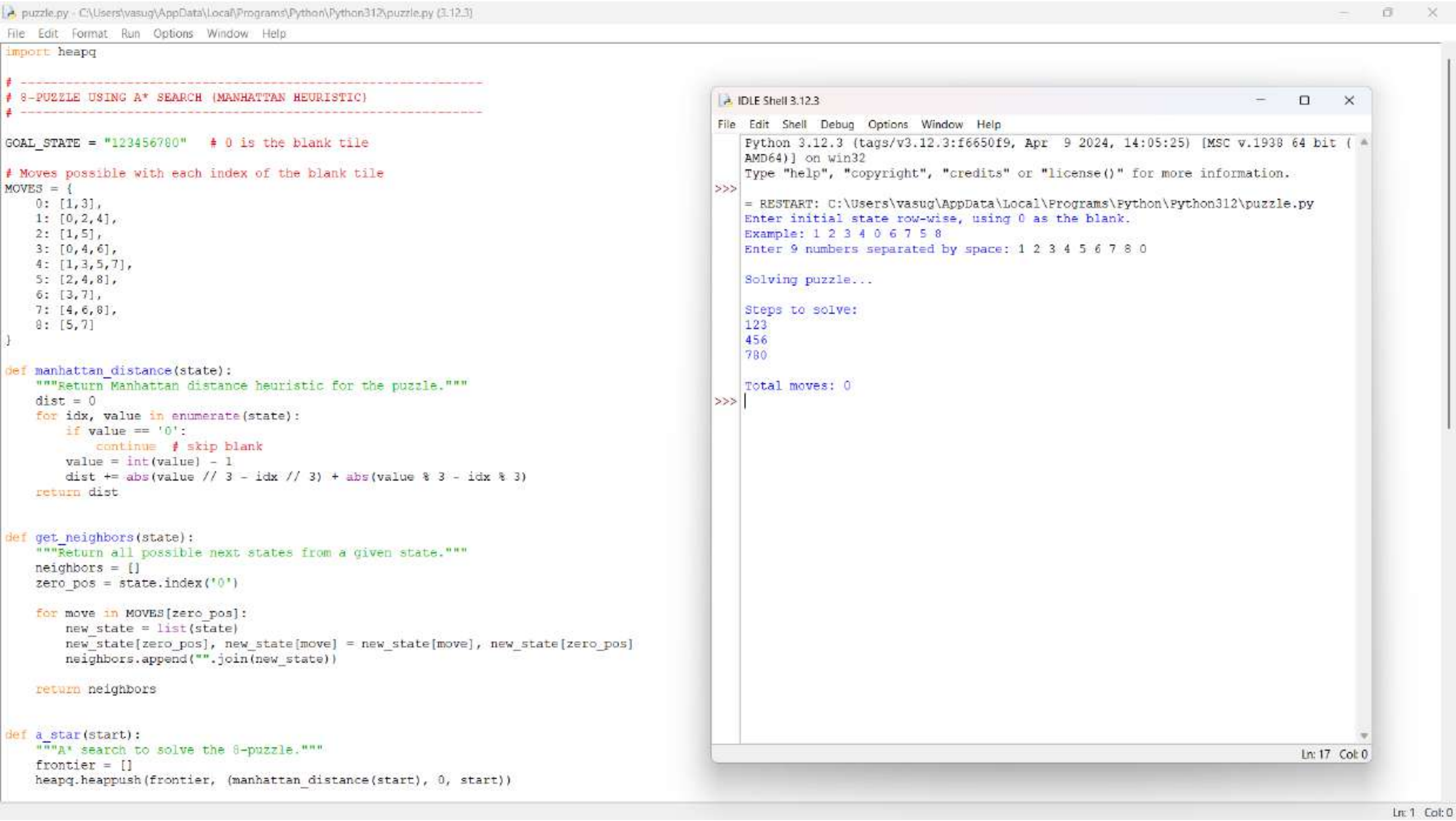
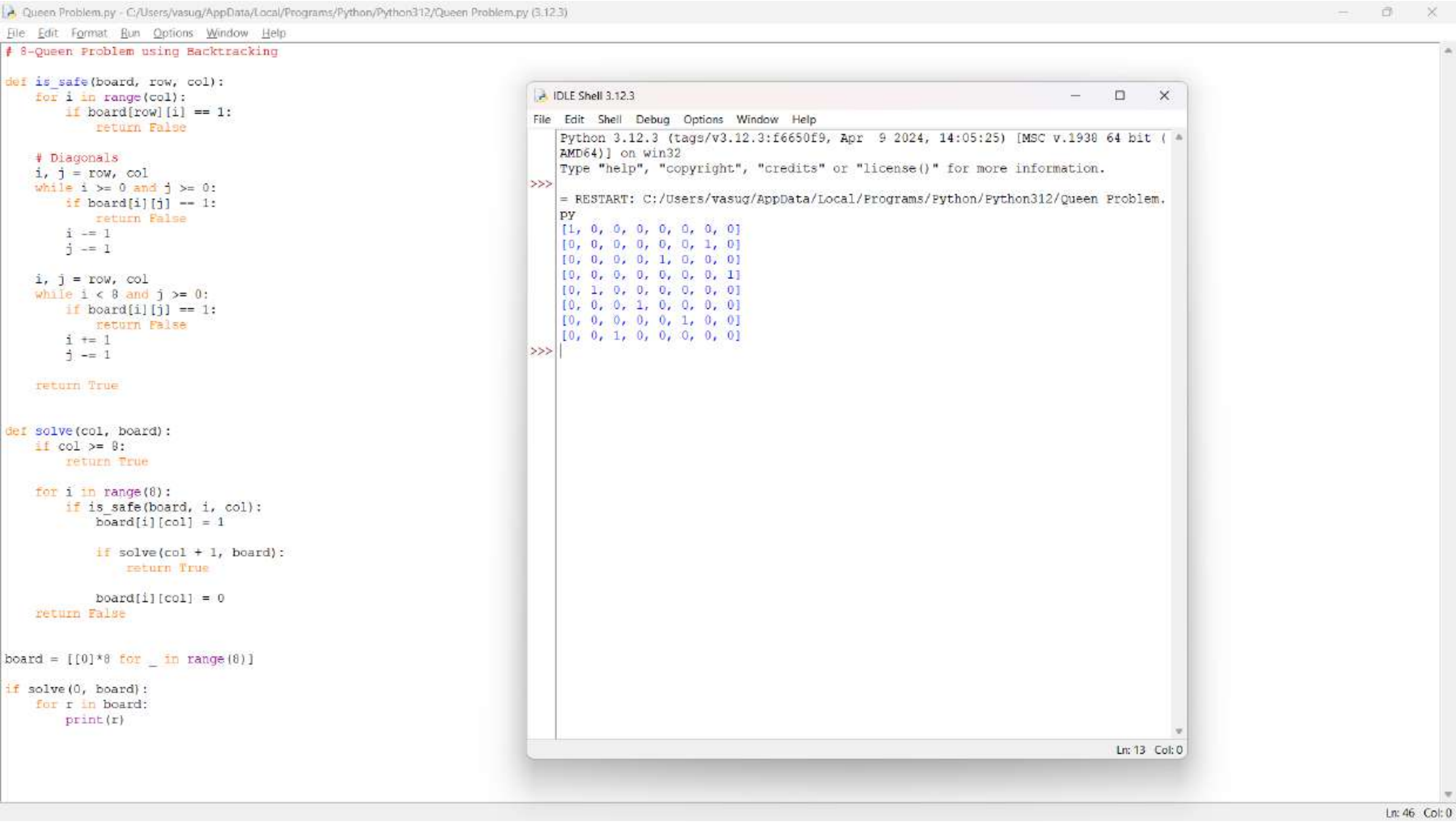

puzzle.py - C:\Users\vasug\AppData\Local\Programs\Python\Python312\puzzle.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 17 Col: 0
Ln: 1 Col: 0

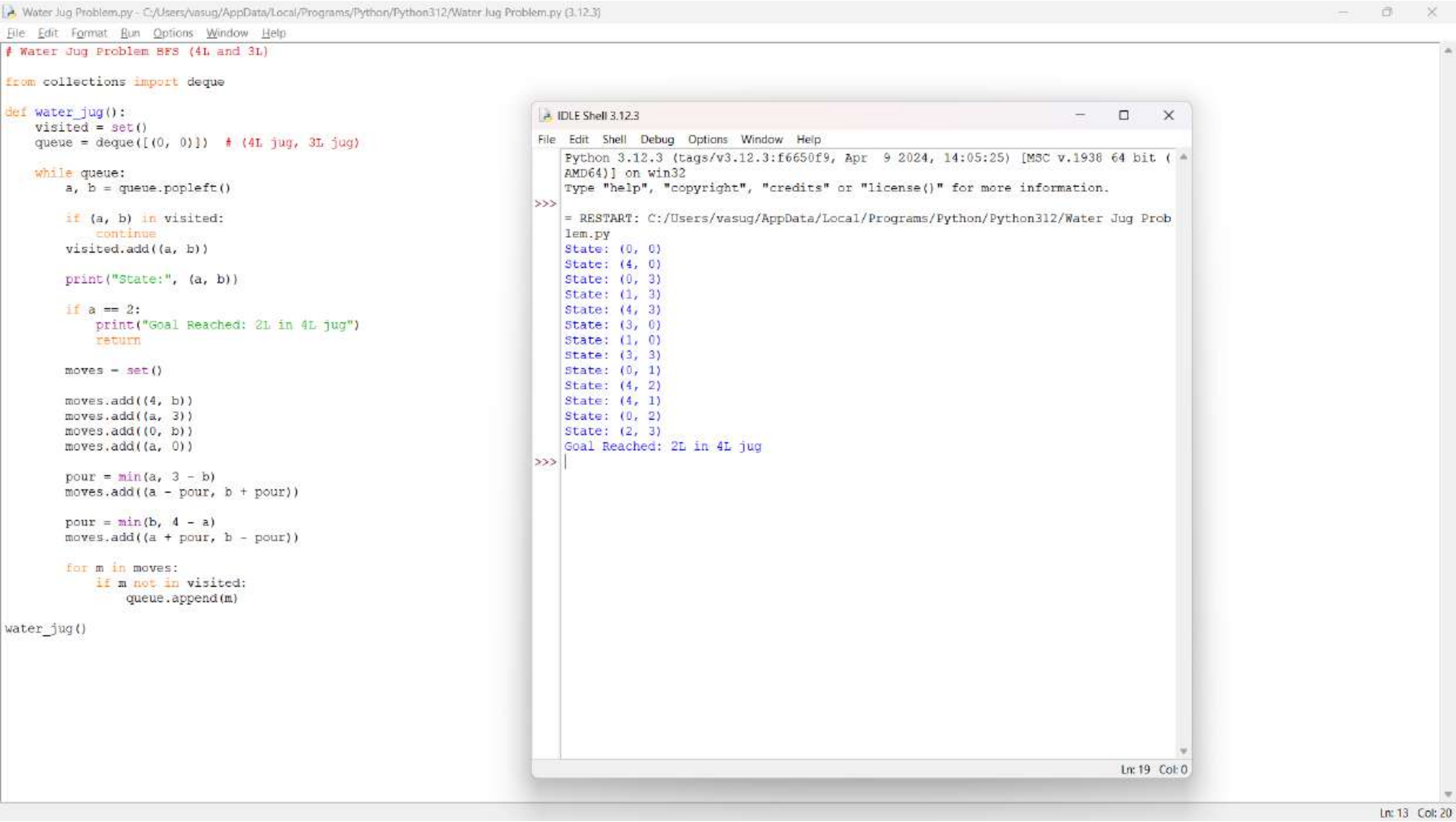
```
import heapq

# ---------------------------------------------------------------
# 8-PUZZLE USING A* SEARCH (MANHATTAN HEURISTIC)
# ---------------------------------------------------------------

GOAL_STATE = "123456780"   # 0 is the blank tile

# Moves possible with each index of the blank tile
MOVES = {
    0: [1,3],
    1: [0,2,4],
    2: [1,5],
    3: [0,4,6],
    4: [1,3,5,7],
    5: [2,4,8],
    6: [3,7],
    7: [4,6,8],
    8: [5,7]
}

def manhattan_distance(state):
    """Return Manhattan distance heuristic for the puzzle."""
    dist = 0
    for idx, value in enumerate(state):
        if value == '0':
            continue  # skip blank
        value = int(value) - 1
        dist += abs(value // 3 - idx // 3) + abs(value % 3 - idx % 3)
    return dist


def get_neighbors(state):
    """Return all possible next states from a given state."""
    neighbors = []
    zero_pos = state.index('0')

    for move in MOVES[zero_pos]:
        new_state = list(state)
        new_state[zero_pos], new_state[move] = new_state[move], new_state[zero_pos]
        neighbors.append("".join(new_state))

    return neighbors


def a_star(start):
    """A* search to solve the 8-puzzle."""
    frontier = []
    heapq.heappush(frontier, (manhattan_distance(start), 0, start))
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:\Users\vasug\AppData\Local\Programs\Python\Python312\puzzle.py
Enter initial state row-wise, using 0 as the blank.
Example: 1 2 3 4 0 6 7 5 8
Enter 9 numbers separated by space: 1 2 3 4 5 6 7 8 0

Solving puzzle...

Steps to solve:
123
456
780

Total moves: 0
>>> 
```


Queen Problem.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Queen Problem.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 13 Col: 0
Ln: 46 Col: 0

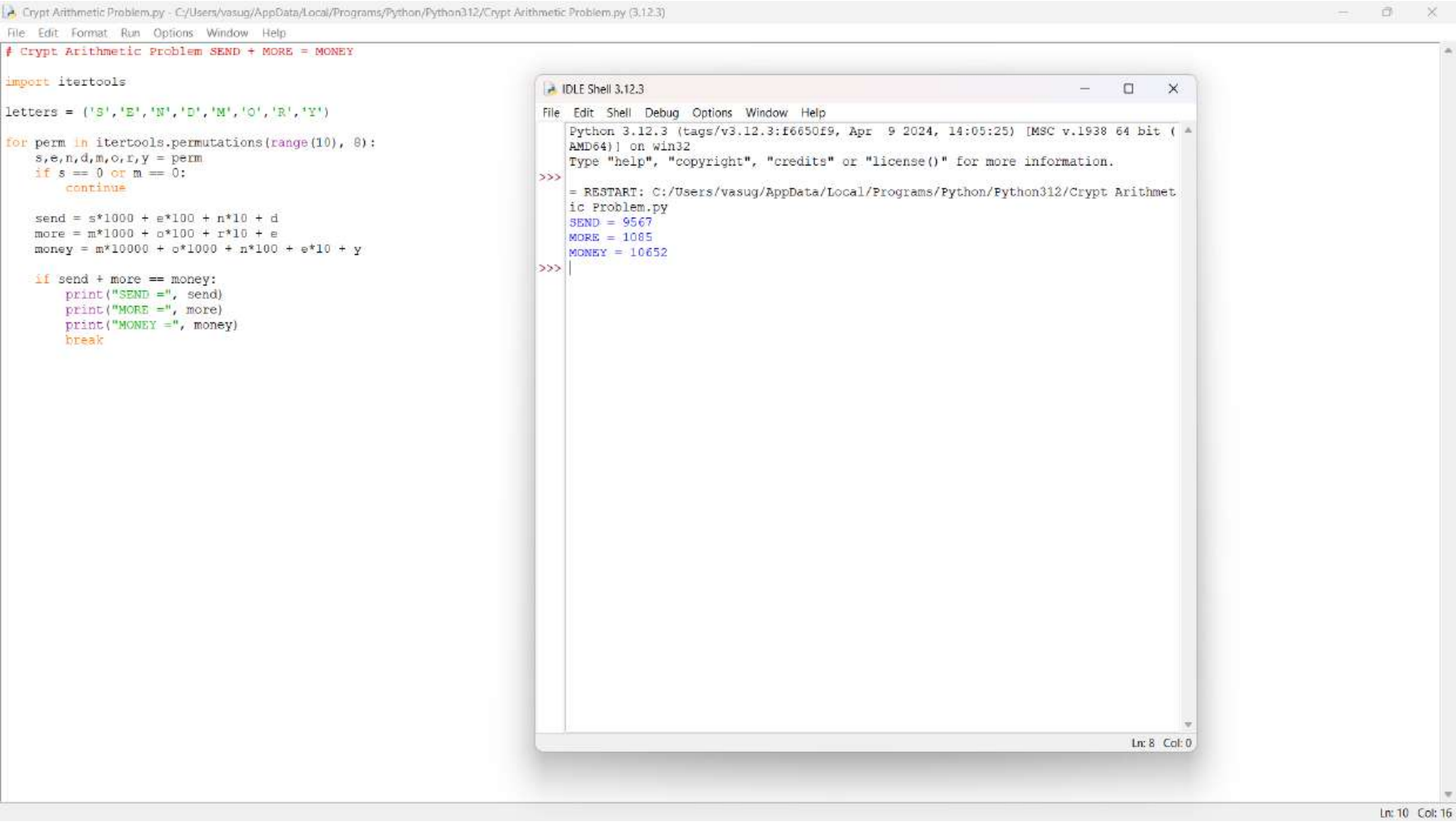
```python
# 8-Queen Problem using Backtracking

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

    # Diagonals
    i, j = row, col
    while i >= 0 and j >= 0:
        if board[i][j] == 1:
            return False
        i -= 1
        j -= 1

    i, j = row, col
    while i < 8 and j >= 0:
        if board[i][j] == 1:
            return False
        i += 1
        j -= 1

    return True


def solve(col, board):
    if col >= 8:
        return True

    for i in range(8):
        if is_safe(board, i, col):
            board[i][col] = 1

            if solve(col + 1, board):
                return True

            board[i][col] = 0
    return False


board = [[0]*8 for _ in range(8)]

if solve(0, board):
    for r in board:
        print(r)
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Queen Problem.
py
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]
[0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0]
>>>
```


Water Jug Problem.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Water Jug Problem.py (3.12.3)
File Edit Format Run Options Window Help
Ln: 13 Col: 20
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 19 Col: 0

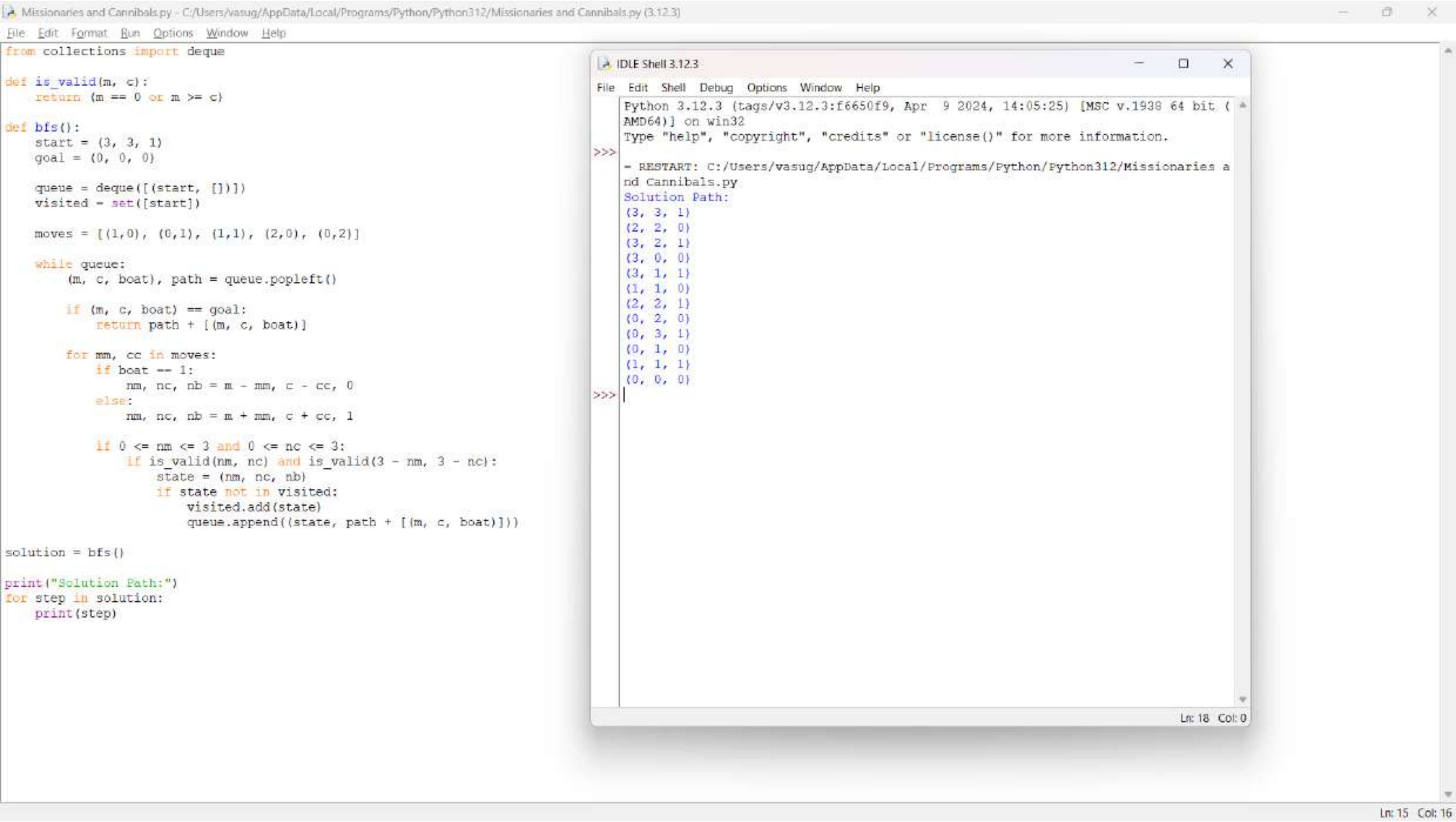
```python
# Water Jug Problem BFS (4L and 3L)

from collections import deque

def water_jug():
    visited = set()
    queue = deque([(0, 0)])  # (4L jug, 3L jug)

    while queue:
        a, b = queue.popleft()

        if (a, b) in visited:
            continue
        visited.add((a, b))

        print("State:", (a, b))

        if a == 2:
            print("Goal Reached: 2L in 4L jug")
            return

        moves = set()

        moves.add((4, b))
        moves.add((a, 3))
        moves.add((0, b))
        moves.add((a, 0))

        pour = min(a, 3 - b)
        moves.add((a - pour, b + pour))

        pour = min(b, 4 - a)
        moves.add((a + pour, b - pour))

        for m in moves:
            if m not in visited:
                queue.append(m)

water_jug()
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Water Jug Prob
lem.py
State: (0, 0)
State: (4, 0)
State: (0, 3)
State: (1, 3)
State: (4, 3)
State: (3, 0)
State: (1, 0)
State: (3, 3)
State: (0, 1)
State: (4, 2)
State: (4, 1)
State: (0, 2)
State: (2, 3)
Goal Reached: 2L in 4L jug
>>>
```


Crypt Arithmetic Problem.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Crypt Arithmetic Problem.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 8 Col: 0
Ln: 10 Col: 16

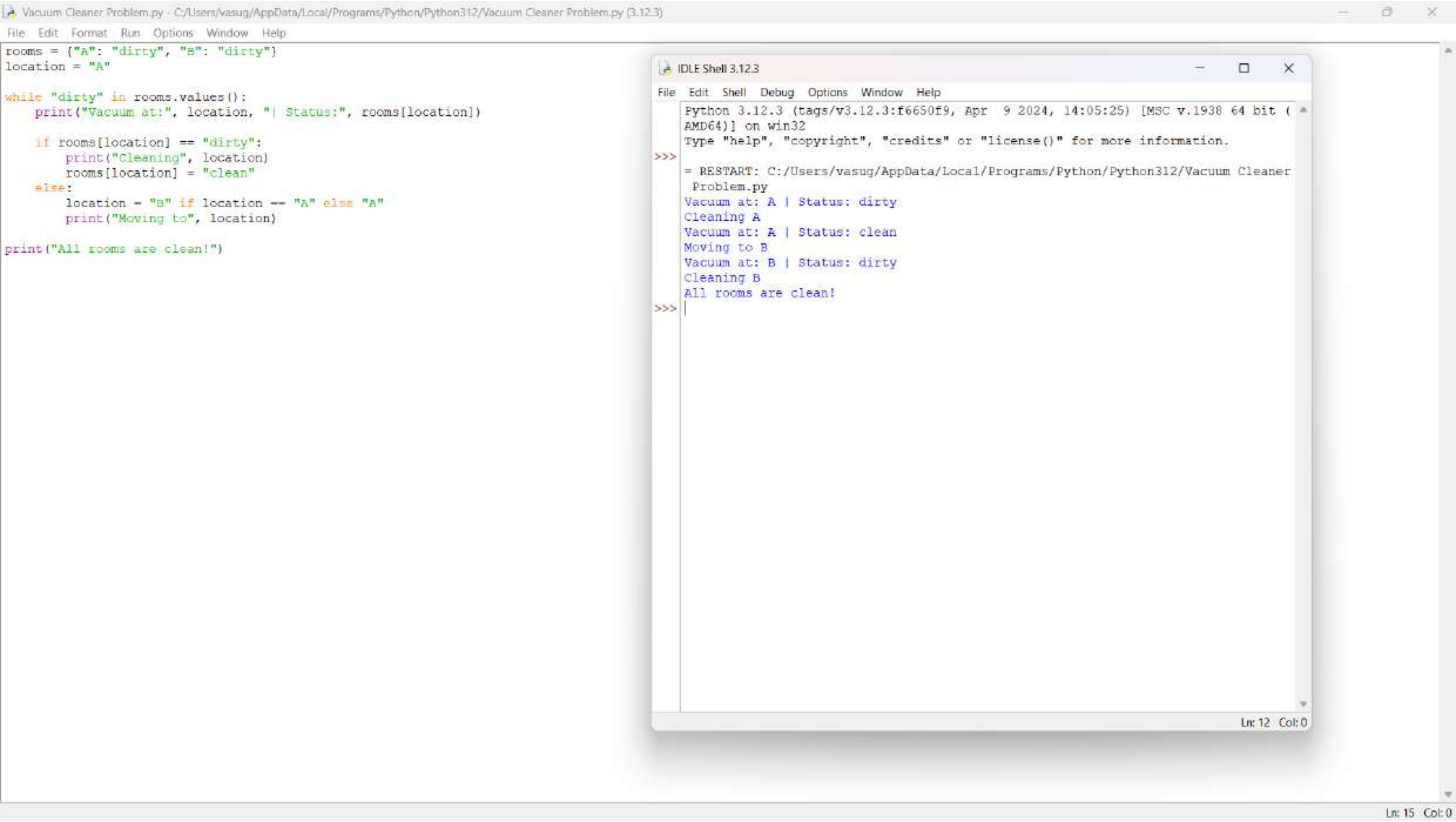
```
# Crypt Arithmetic Problem SEND + MORE = MONEY

import itertools

letters = ('S','E','N','D','M','O','R','Y')

for perm in itertools.permutations(range(10), 8):
    s,e,n,d,m,o,r,y = perm
    if s == 0 or m == 0:
        continue

    send = s*1000 + e*100 + n*10 + d
    more = m*1000 + o*100 + r*10 + e
    money = m*10000 + o*1000 + n*100 + e*10 + y

    if send + more == money:
        print("SEND =", send)
        print("MORE =", more)
        print("MONEY =", money)
        break
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Crypt Arithmet
ic Problem.py
SEND = 9567
MORE = 1085
MONEY = 10652
>>> 
```


Missionaries and Cannibals.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Missionaries and Cannibals.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 18 Col: 0
Ln: 15 Col: 16

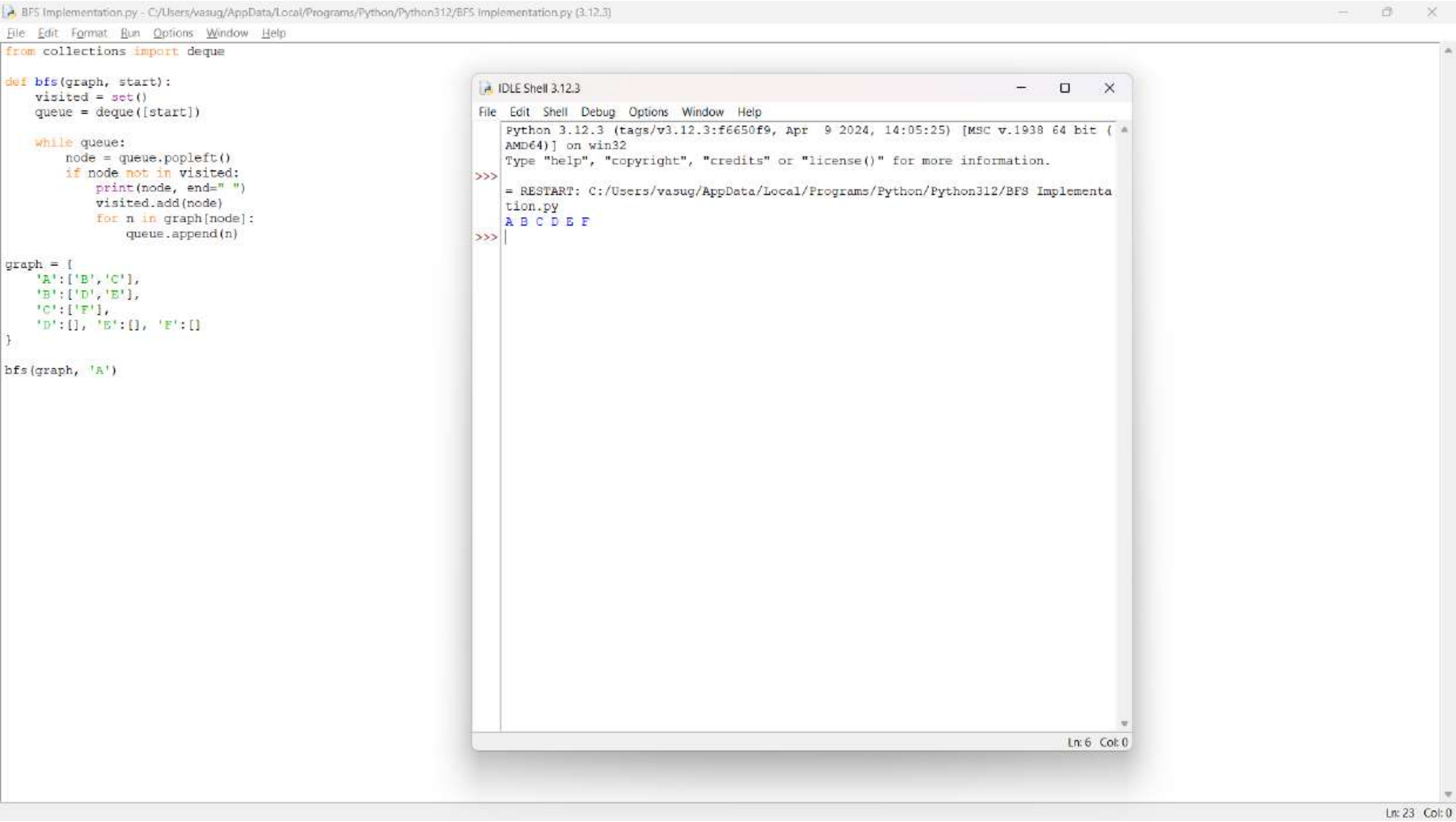
```python
from collections import deque

def is_valid(m, c):
    return (m == 0 or m >= c)

def bfs():
    start = (3, 3, 1)
    goal = (0, 0, 0)

    queue = deque([(start, [])])
    visited = set([start])

    moves = [(1,0), (0,1), (1,1), (2,0), (0,2)]

    while queue:
        (m, c, boat), path = queue.popleft()

        if (m, c, boat) == goal:
            return path + [(m, c, boat)]

        for mm, cc in moves:
            if boat == 1:
                nm, nc, nb = m - mm, c - cc, 0
            else:
                nm, nc, nb = m + mm, c + cc, 1

            if 0 <= nm <= 3 and 0 <= nc <= 3:
                if is_valid(nm, nc) and is_valid(3 - nm, 3 - nc):
                    state = (nm, nc, nb)
                    if state not in visited:
                        visited.add(state)
                        queue.append((state, path + [(m, c, boat)]))

solution = bfs()

print("Solution Path:")
for step in solution:
    print(step)
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Missionaries a
nd Cannibals.py
Solution Path:
(3, 3, 1)
(2, 2, 0)
(3, 2, 1)
(3, 0, 0)
(3, 1, 1)
(1, 1, 0)
(2, 2, 1)
(0, 2, 0)
(0, 3, 1)
(0, 1, 0)
(1, 1, 1)
(0, 0, 0)
>>> 
```


Vacuum Cleaner Problem.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Vacuum Cleaner Problem.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 12 Col: 0
Ln: 15 Col: 0

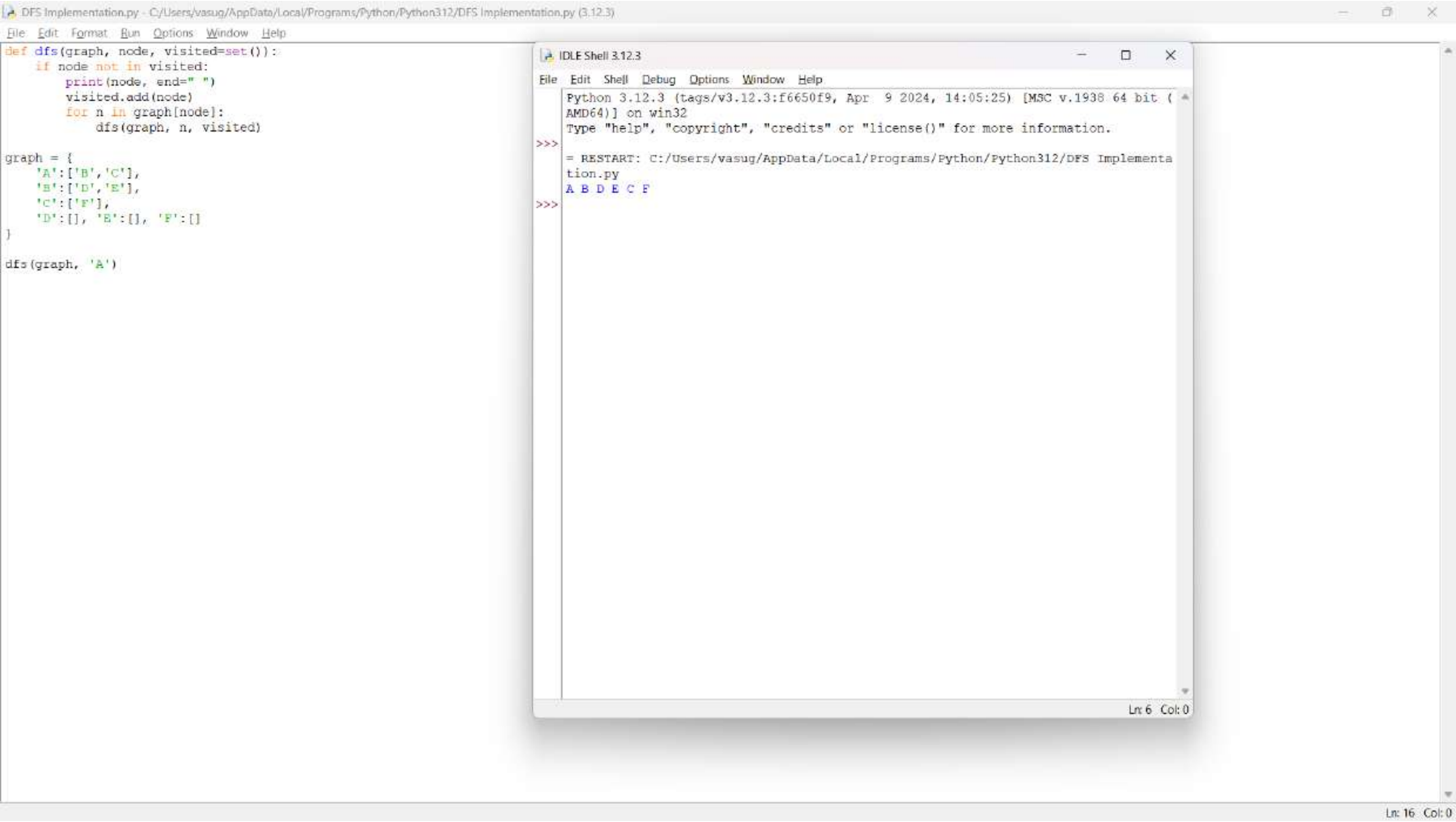
```python
rooms = {"A": "dirty", "B": "dirty"}
location = "A"

while "dirty" in rooms.values():
    print("Vacuum at:", location, "| Status:", rooms[location])

    if rooms[location] == "dirty":
        print("Cleaning", location)
        rooms[location] = "clean"
    else:
        location = "B" if location == "A" else "A"
        print("Moving to", location)

print("All rooms are clean!")
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Vacuum Cleaner
 Problem.py
Vacuum at: A | Status: dirty
Cleaning A
Vacuum at: A | Status: clean
Moving to B
Vacuum at: B | Status: dirty
Cleaning B
All rooms are clean!
>>>
```


BFS Implementation.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/BFS Implementation.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 23 Col: 0

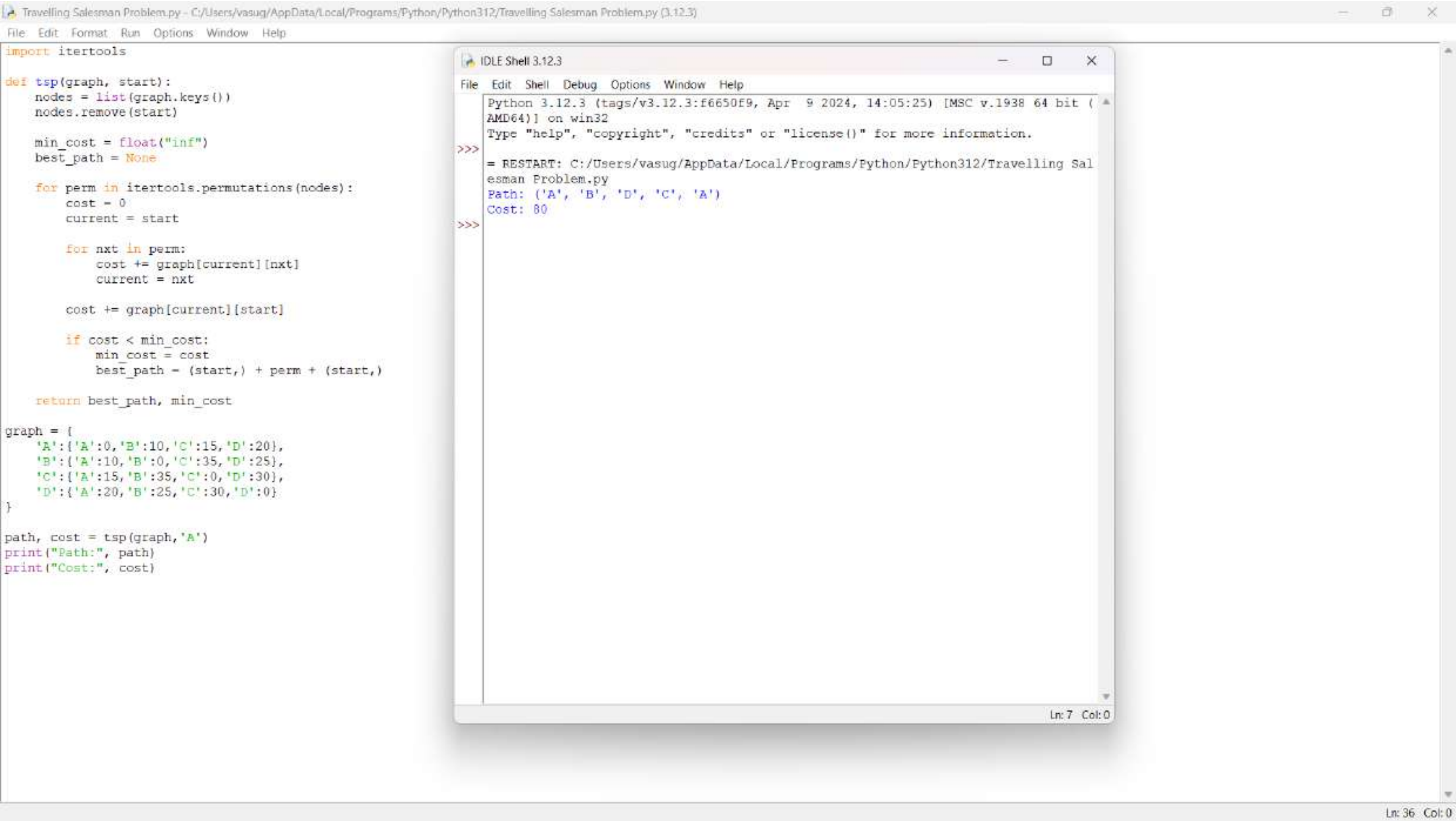
```python
from collections import deque

def bfs(graph, start):
    visited = set()
    queue = deque([start])

    while queue:
        node = queue.popleft()
        if node not in visited:
            print(node, end=" ")
            visited.add(node)
            for n in graph[node]:
                queue.append(n)

graph = {
    'A':['B','C'],
    'B':['D','E'],
    'C':['F'],
    'D':[], 'E':[], 'F':[]
}

bfs(graph, 'A')
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/BFS Implementa
tion.py
A B C D E F
>>>
```


DFS Implementation.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/DFS Implementation.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 16 Col: 0

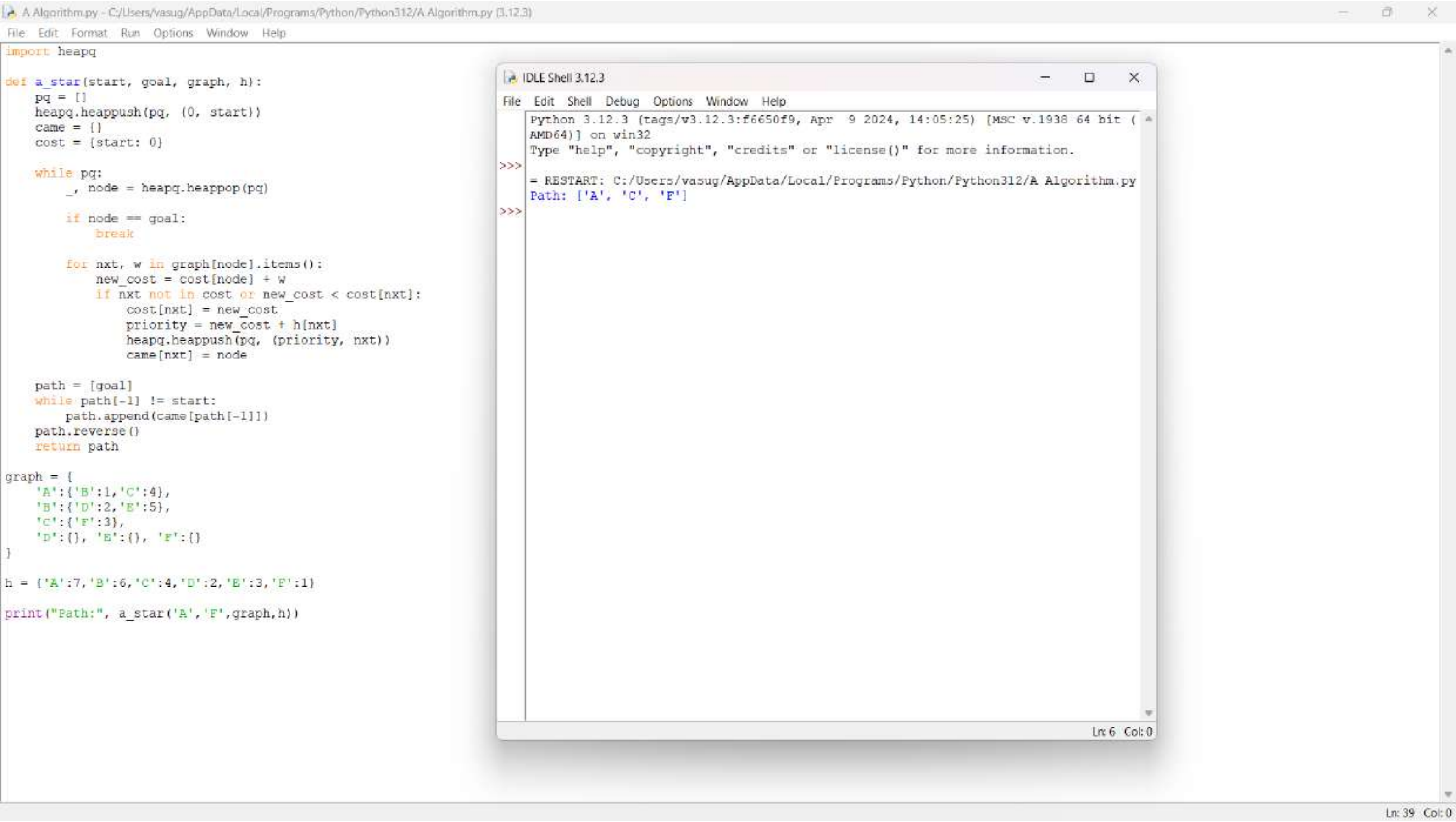
```
def dfs(graph, node, visited=set()):
    if node not in visited:
        print(node, end=" ")
        visited.add(node)
        for n in graph[node]:
            dfs(graph, n, visited)

graph = {
    'A':['B','C'],
    'B':['D','E'],
    'C':['F'],
    'D':[], 'E':[], 'F':[]
}

dfs(graph, 'A')
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/DFS Implementa
tion.py
A B D E C F
>>>
```


Travelling Salesman Problem.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Travelling Salesman Problem.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 7 Col: 0
Ln: 36 Col: 0

```python
import itertools

def tsp(graph, start):
    nodes = list(graph.keys())
    nodes.remove(start)

    min_cost = float("inf")
    best_path = None

    for perm in itertools.permutations(nodes):
        cost = 0
        current = start

        for nxt in perm:
            cost += graph[current][nxt]
            current = nxt

        cost += graph[current][start]

        if cost < min_cost:
            min_cost = cost
            best_path = (start,) + perm + (start,)

    return best_path, min_cost

graph = {
    'A':{'A':0,'B':10,'C':15,'D':20},
    'B':{'A':10,'B':0,'C':35,'D':25},
    'C':{'A':15,'B':35,'C':0,'D':30},
    'D':{'A':20,'B':25,'C':30,'D':0}
}

path, cost = tsp(graph,'A')
print("Path:", path)
print("Cost:", cost)
```

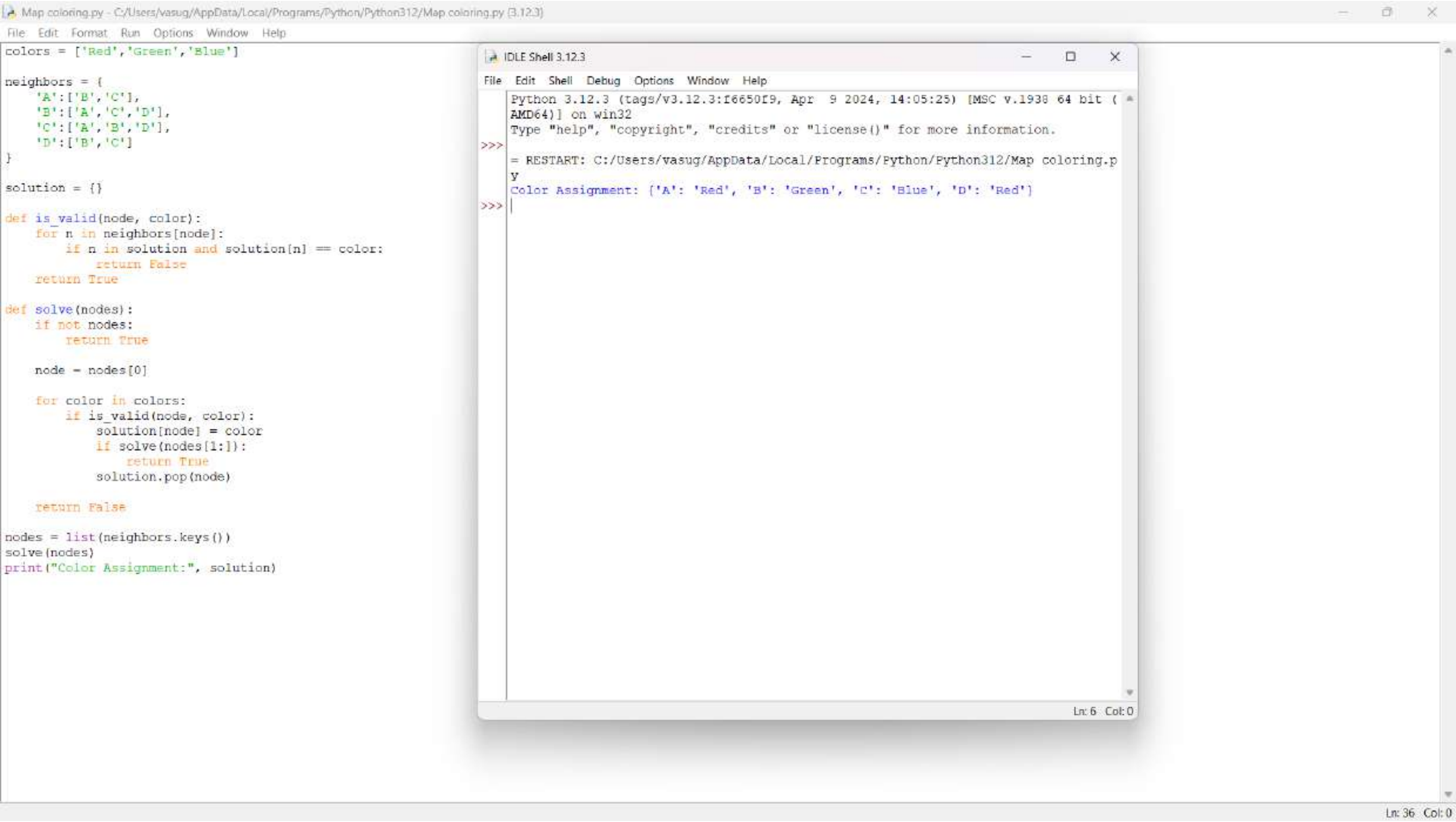
```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Travelling Sal
esman Problem.py
Path: ('A', 'B', 'D', 'C', 'A')
Cost: 80
>>> 
```



A Algorithm.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/A Algorithm.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 39 Col: 0

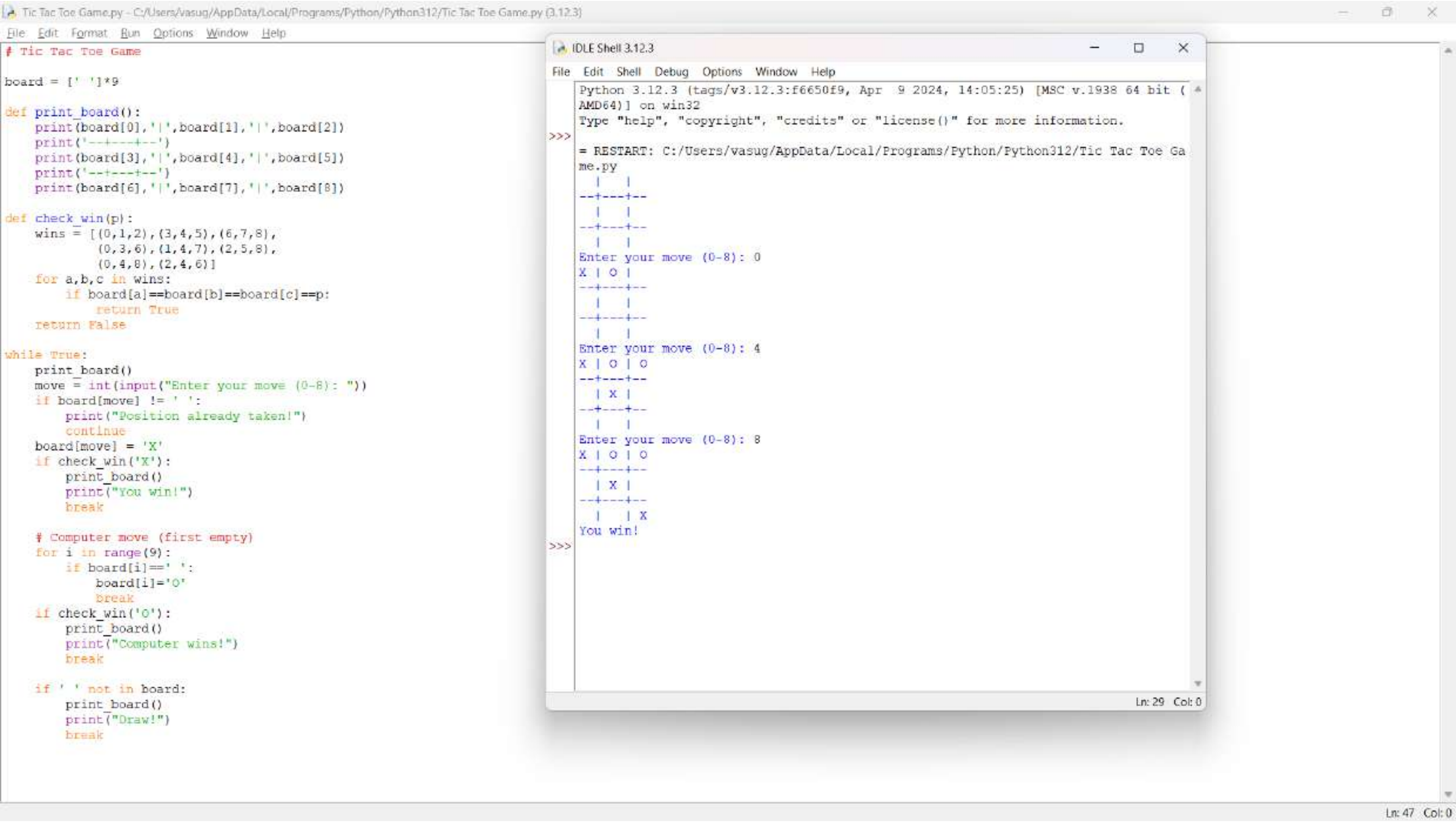
```python
import heapq

def a_star(start, goal, graph, h):
    pq = []
    heapq.heappush(pq, (0, start))
    came = {}
    cost = {start: 0}

    while pq:
        _, node = heapq.heappop(pq)

        if node == goal:
            break

        for nxt, w in graph[node].items():
            new_cost = cost[node] + w
            if nxt not in cost or new_cost < cost[nxt]:
                cost[nxt] = new_cost
                priority = new_cost + h[nxt]
                heapq.heappush(pq, (priority, nxt))
                came[nxt] = node

    path = [goal]
    while path[-1] != start:
        path.append(came[path[-1]])
    path.reverse()
    return path

graph = {
    'A':{'B':1,'C':4},
    'B':{'D':2,'E':5},
    'C':{'F':3},
    'D':{}, 'E':{}, 'F':{}
}

h = {'A':7,'B':6,'C':4,'D':2,'E':3,'F':1}

print("Path:", a_star('A','F',graph,h))
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/A Algorithm.py
Path: ['A', 'C', 'F']
>>>
```


Map coloring.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Map coloring.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 36 Col: 0

```python
colors = ['Red','Green','Blue']

neighbors = {
    'A':['B','C'],
    'B':['A','C','D'],
    'C':['A','B','D'],
    'D':['B','C']
}

solution = {}

def is_valid(node, color):
    for n in neighbors[node]:
        if n in solution and solution[n] == color:
            return False
    return True

def solve(nodes):
    if not nodes:
        return True

    node = nodes[0]

    for color in colors:
        if is_valid(node, color):
            solution[node] = color
            if solve(nodes[1:]):
                return True
            solution.pop(node)

    return False

nodes = list(neighbors.keys())
solve(nodes)
print("Color Assignment:", solution)
```

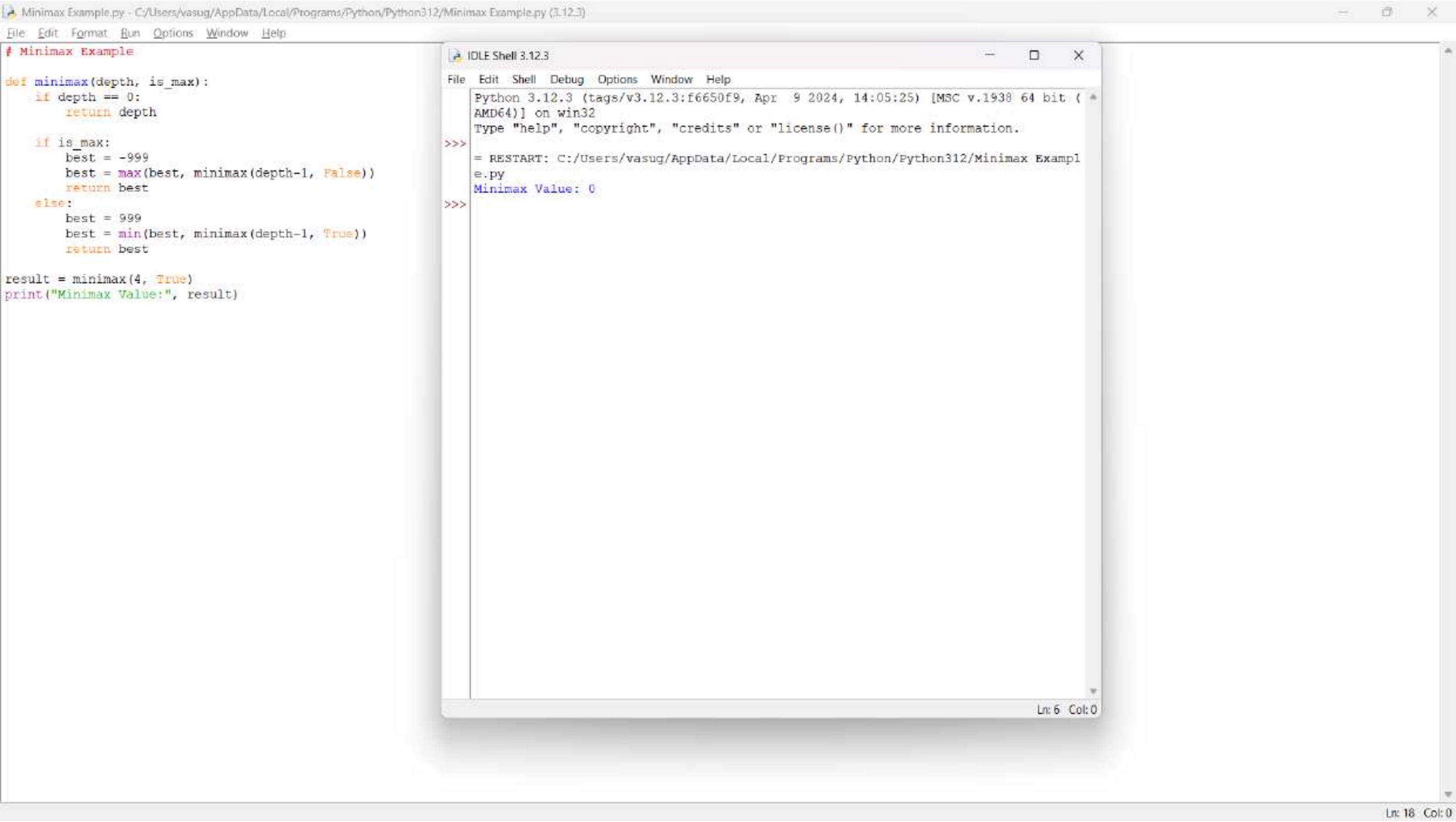
```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Map coloring.p
y
Color Assignment: {'A': 'Red', 'B': 'Green', 'C': 'Blue', 'D': 'Red'}
>>>
```



Tic Tac Toe Game.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Tic Tac Toe Game.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 29 Col: 0
Ln: 47 Col: 0

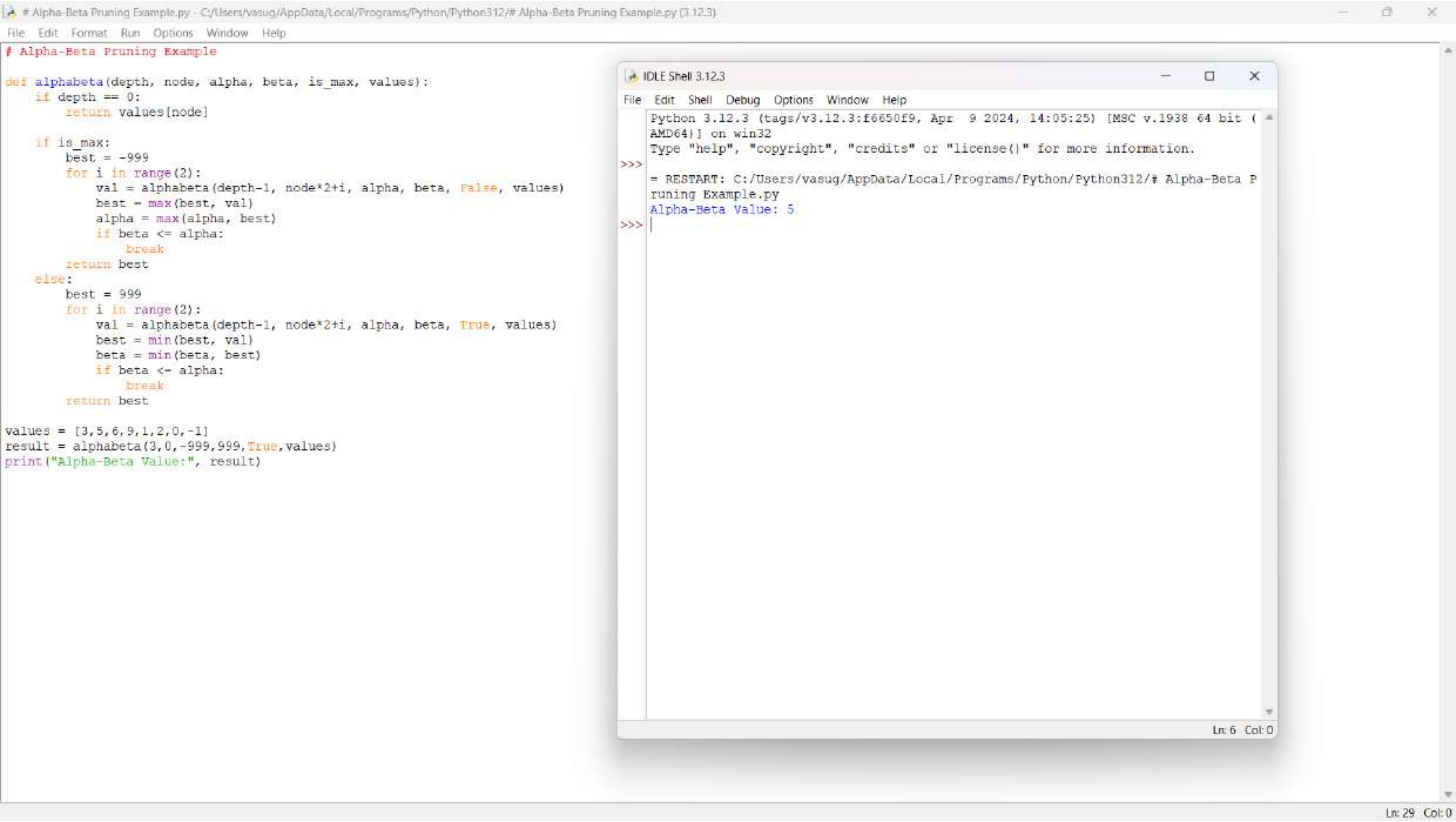
```python
# Tic Tac Toe Game

board = [' ']*9

def print_board():
    print(board[0],'|',board[1],'|',board[2])
    print('--+---+--')
    print(board[3],'|',board[4],'|',board[5])
    print('--+---+--')
    print(board[6],'|',board[7],'|',board[8])

def check_win(p):
    wins = [(0,1,2),(3,4,5),(6,7,8),
            (0,3,6),(1,4,7),(2,5,8),
            (0,4,8),(2,4,6)]
    for a,b,c in wins:
        if board[a]==board[b]==board[c]==p:
            return True
    return False

while True:
    print_board()
    move = int(input("Enter your move (0-8): "))
    if board[move] != ' ':
        print("Position already taken!")
        continue
    board[move] = 'X'
    if check_win('X'):
        print_board()
        print("You win!")
        break

    # Computer move (first empty)
    for i in range(9):
        if board[i]==' ':
            board[i]='O'
            break
    if check_win('O'):
        print_board()
        print("Computer wins!")
        break

    if ' ' not in board:
        print_board()
        print("Draw!")
        break
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Tic Tac Toe Ga
me.py
  |   |  
--+---+--
  |   |  
--+---+--
  |   |  
Enter your move (0-8): 0
X | O |  
--+---+--
  |   |  
--+---+--
  |   |  
Enter your move (0-8): 4
X | O | O
--+---+--
  | X |  
--+---+--
  |   |  
Enter your move (0-8): 8
X | O | O
--+---+--
  | X |  
--+---+--
  |   | X
You win!
>>>
```


Minimax Example.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Minimax Example.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 18 Col: 0

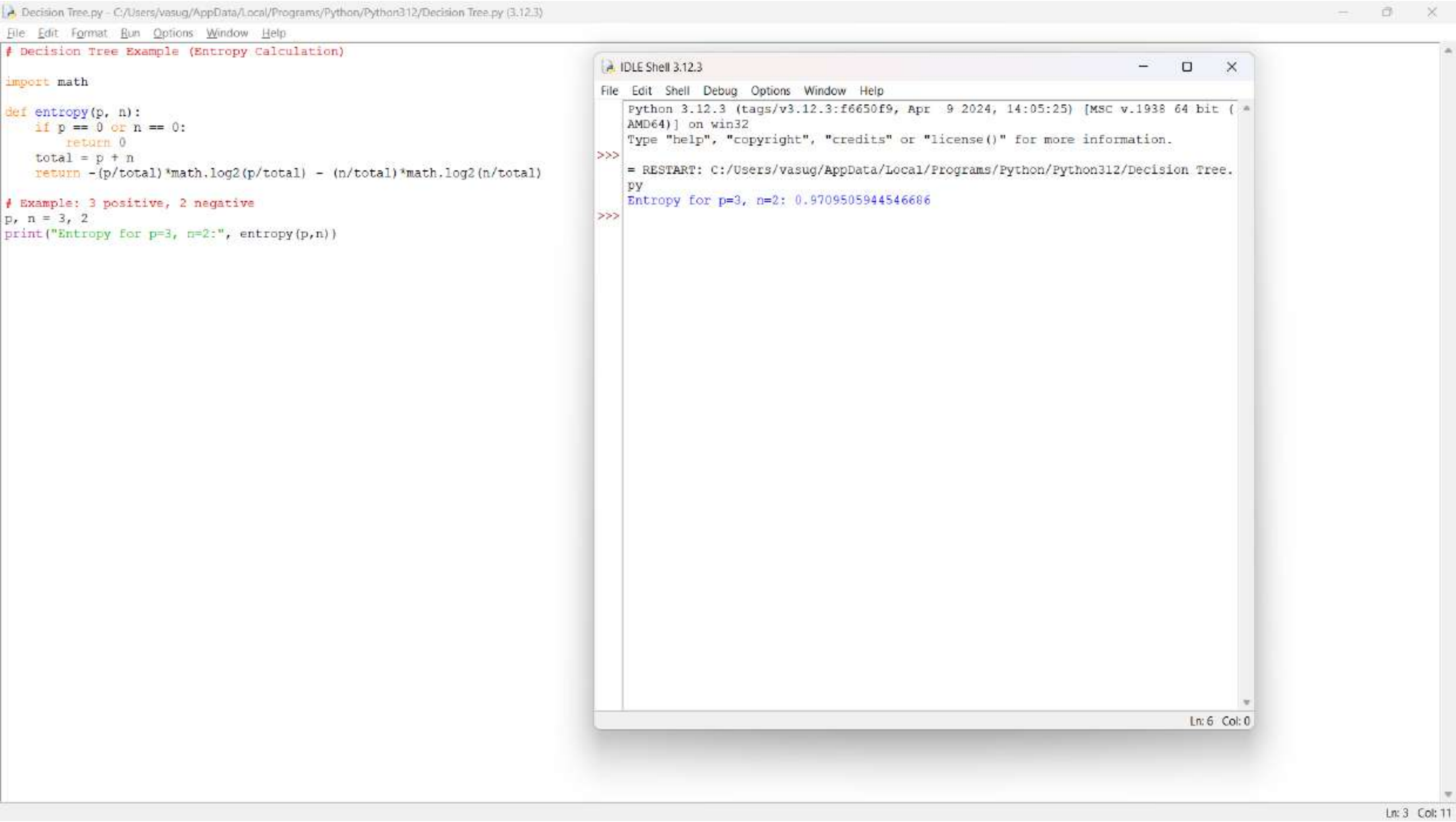
```
# Minimax Example

def minimax(depth, is_max):
    if depth == 0:
        return depth

    if is_max:
        best = -999
        best = max(best, minimax(depth-1, False))
        return best
    else:
        best = 999
        best = min(best, minimax(depth-1, True))
        return best

result = minimax(4, True)
print("Minimax Value:", result)
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Minimax Exampl
e.py
Minimax Value: 0
>>>
```


# Alpha-Beta Pruning Example.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/# Alpha-Beta Pruning Example.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 29 Col: 0

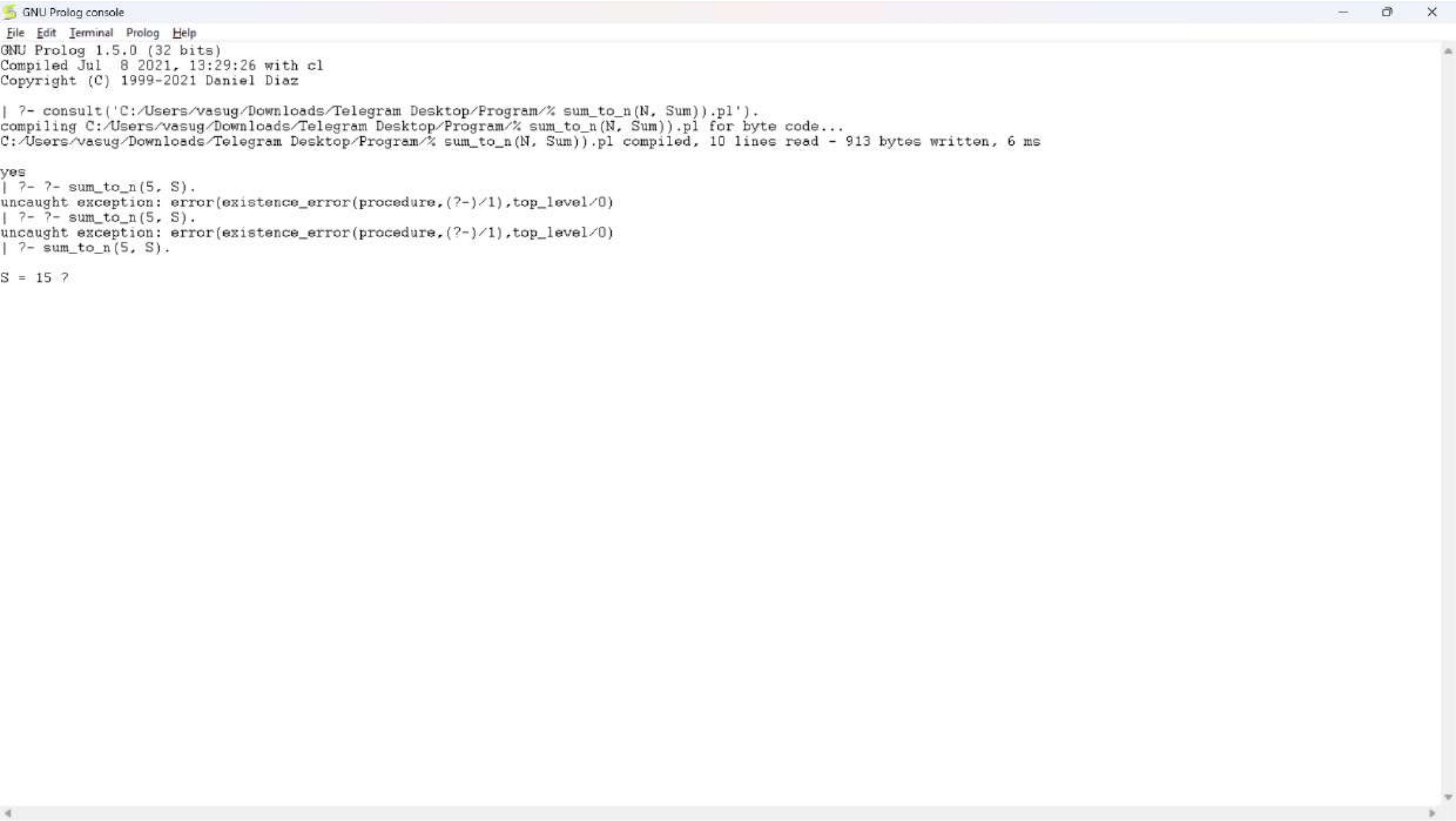
```
# Alpha-Beta Pruning Example

def alphabeta(depth, node, alpha, beta, is_max, values):
    if depth == 0:
        return values[node]

    if is_max:
        best = -999
        for i in range(2):
            val = alphabeta(depth-1, node*2+i, alpha, beta, False, values)
            best = max(best, val)
            alpha = max(alpha, best)
            if beta <= alpha:
                break
        return best
    else:
        best = 999
        for i in range(2):
            val = alphabeta(depth-1, node*2+i, alpha, beta, True, values)
            best = min(best, val)
            beta = min(beta, best)
            if beta <= alpha:
                break
        return best

values = [3,5,6,9,1,2,0,-1]
result = alphabeta(3,0,-999,999,True,values)
print("Alpha-Beta Value:", result)
```

```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/# Alpha-Beta P
runing Example.py
Alpha-Beta Value: 5
>>> 
```

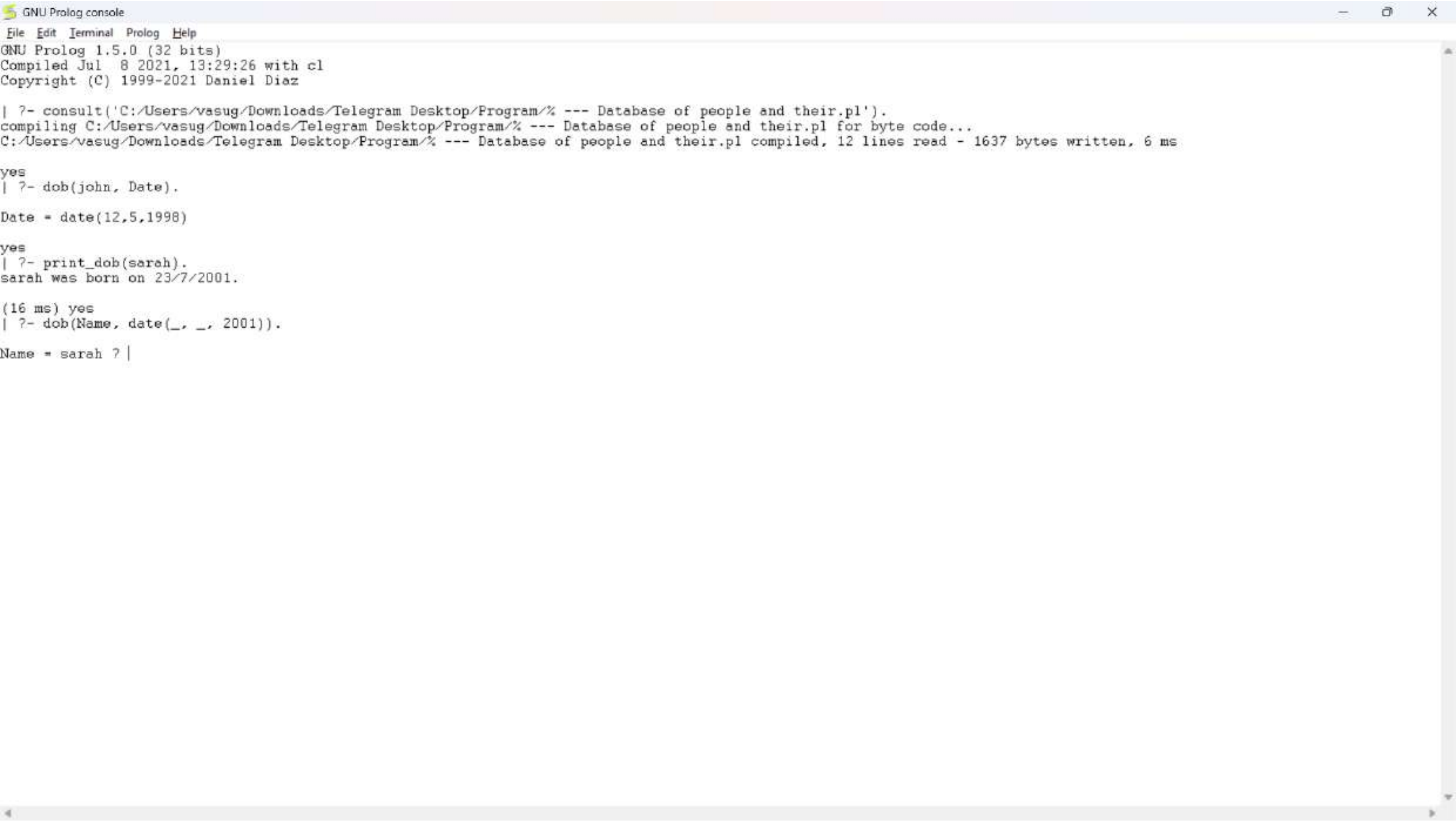

Decision Tree.py - C:/Users/vasug/AppData/Local/Programs/Python/Python312/Decision Tree.py (3.12.3)
File Edit Format Run Options Window Help
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0
Ln: 3 Col: 11

```python
# Decision Tree Example (Entropy Calculation)

import math

def entropy(p, n):
    if p == 0 or n == 0:
        return 0
    total = p + n
    return -(p/total)*math.log2(p/total) - (n/total)*math.log2(n/total)

# Example: 3 positive, 2 negative
p, n = 3, 2
print("Entropy for p=3, n=2:", entropy(p,n))
```

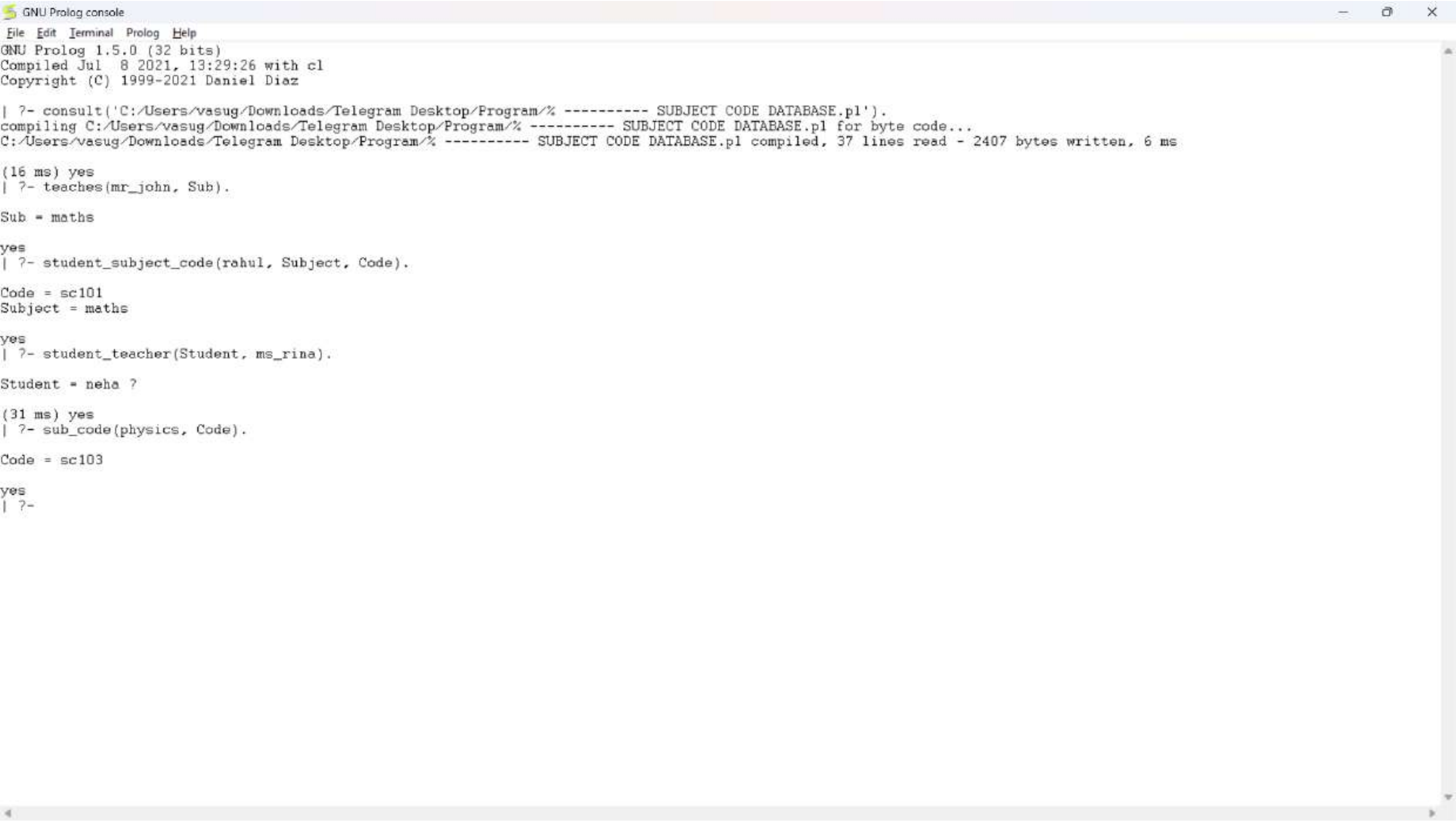
```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr  9 2024, 14:05:25) [MSC v.1938 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/vasug/AppData/Local/Programs/Python/Python312/Decision Tree.
py
Entropy for p=3, n=2: 0.9709505944546686
>>>
```



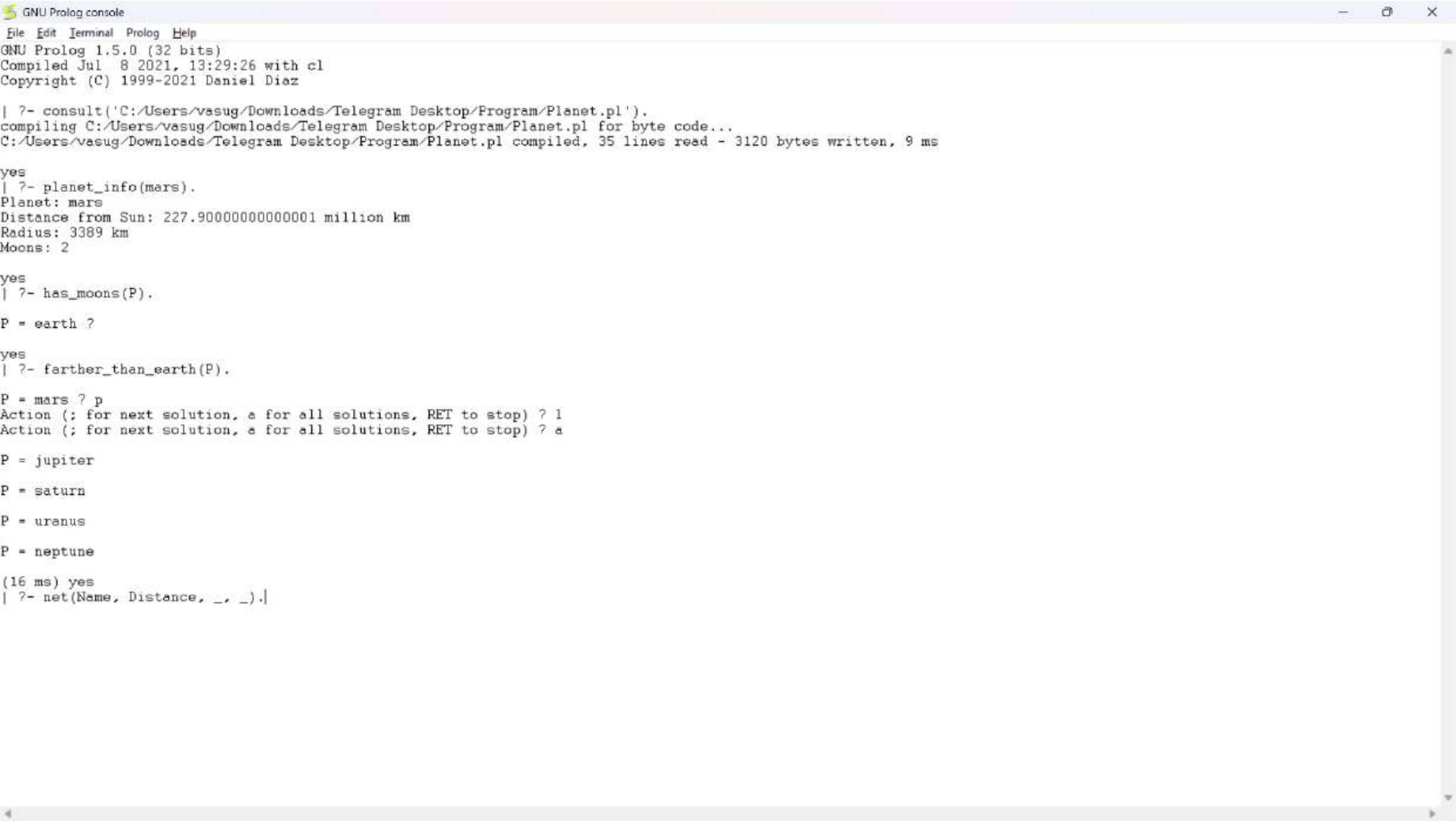
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/% sum_to_n(N, Sum)).pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/% sum_to_n(N, Sum)).pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/% sum_to_n(N, Sum)).pl compiled, 10 lines read - 913 bytes written, 6 ms

yes
| ?- ?- sum_to_n(5, S).
uncaught exception: error(existence_error(procedure,(?-)/1),top_level/0)
| ?- ?- sum_to_n(5, S).
uncaught exception: error(existence_error(procedure,(?-)/1),top_level/0)
| ?- sum_to_n(5, S).

S = 15 ?
```



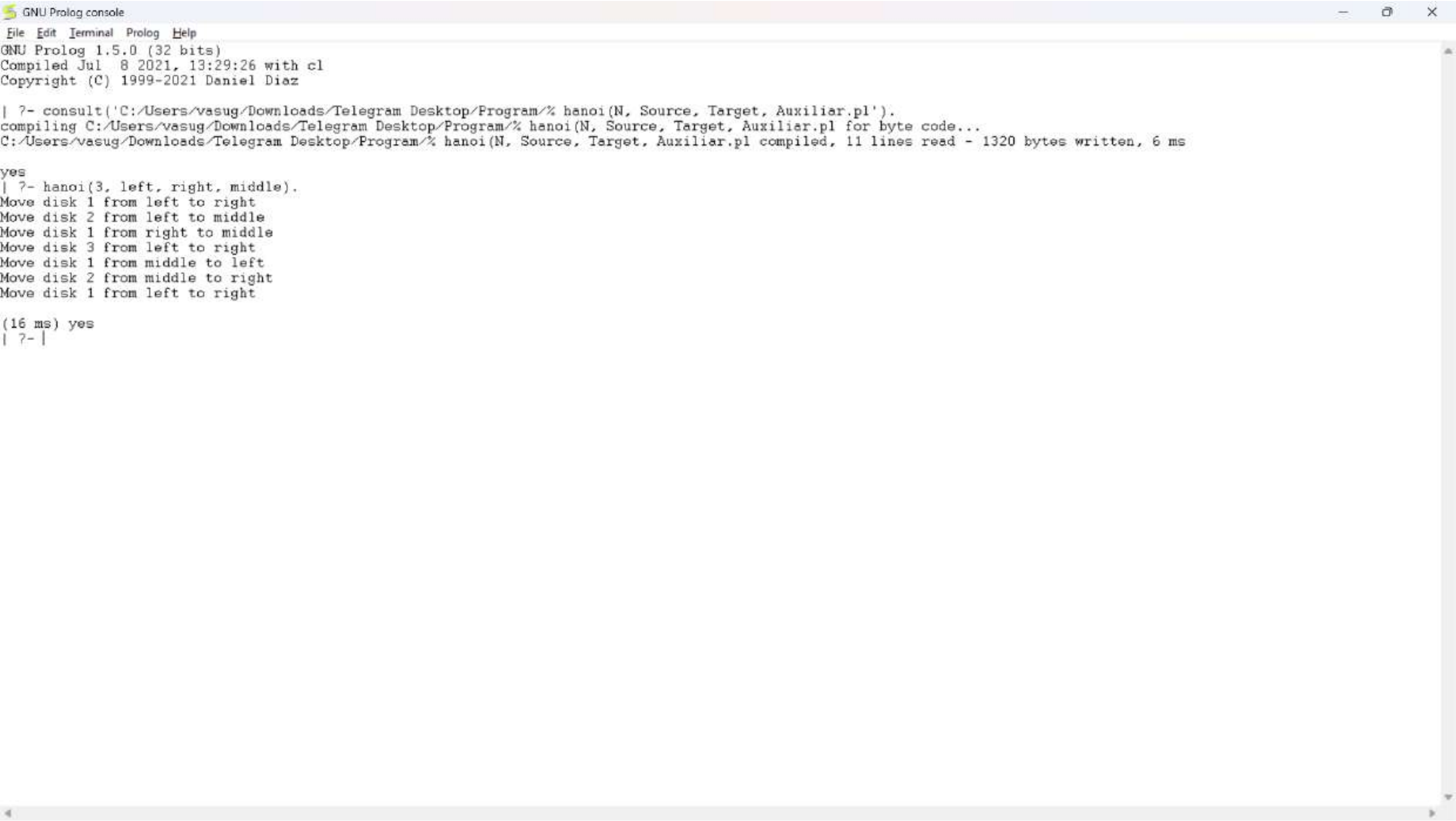
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/% --- Database of people and their.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/% --- Database of people and their.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/% --- Database of people and their.pl compiled, 12 lines read - 1637 bytes written, 6 ms

yes
| ?- dob(john, Date).

Date = date(12,5,1998)

yes
| ?- print_dob(sarah).
sarah was born on 23/7/2001.

(16 ms) yes
| ?- dob(Name, date(_, _, 2001)).

Name = sarah ? |
```



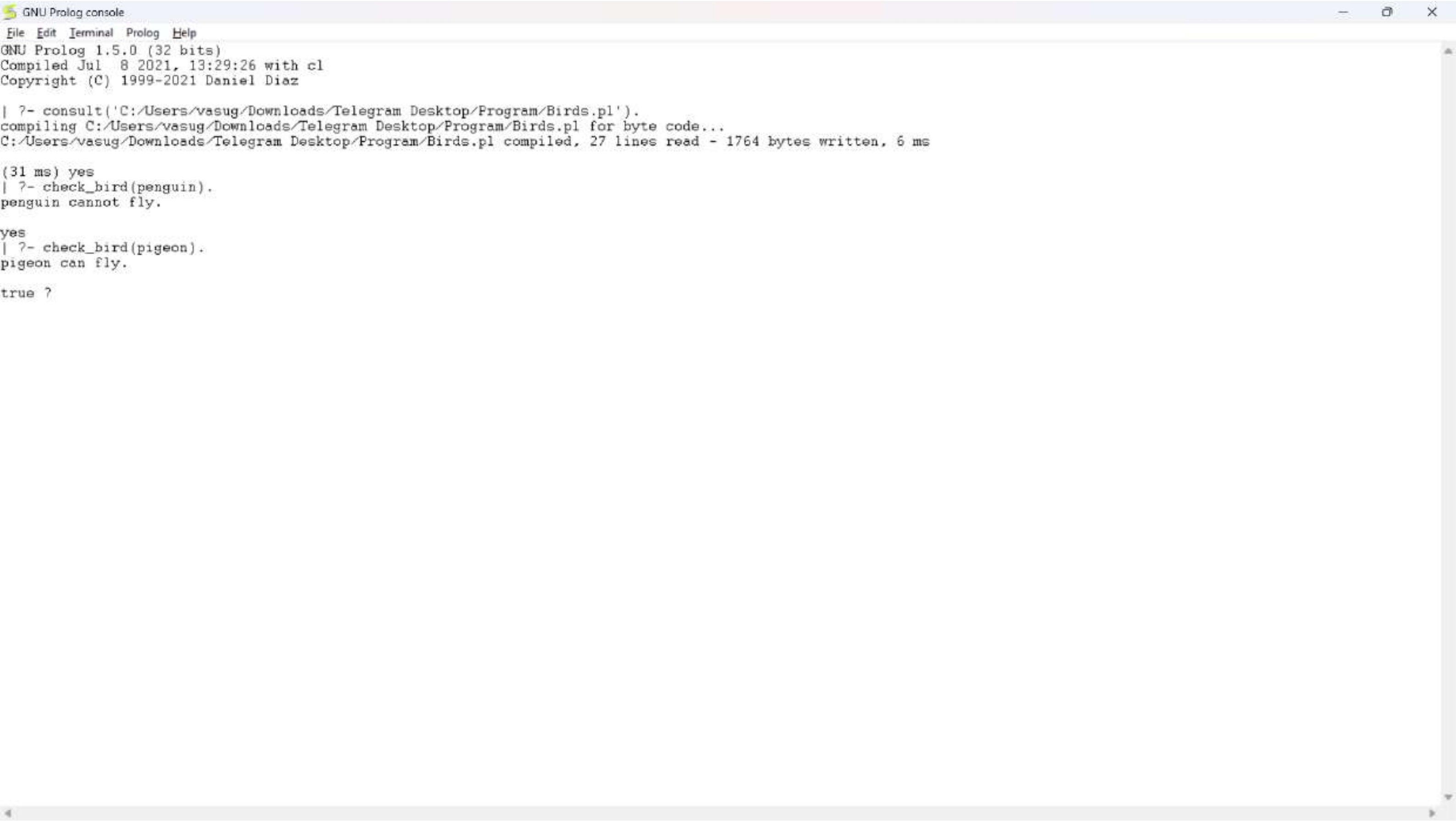
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/% ---------- SUBJECT CODE DATABASE.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/% ---------- SUBJECT CODE DATABASE.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/% ---------- SUBJECT CODE DATABASE.pl compiled, 37 lines read - 2407 bytes written, 6 ms

(16 ms) yes
| ?- teaches(mr_john, Sub).

Sub = maths

yes
| ?- student_subject_code(rahul, Subject, Code).

Code = sc101
Subject = maths

yes
| ?- student_teacher(Student, ms_rina).

Student = neha ?

(31 ms) yes
| ?- sub_code(physics, Code).

Code = sc103

yes
| ?-
```



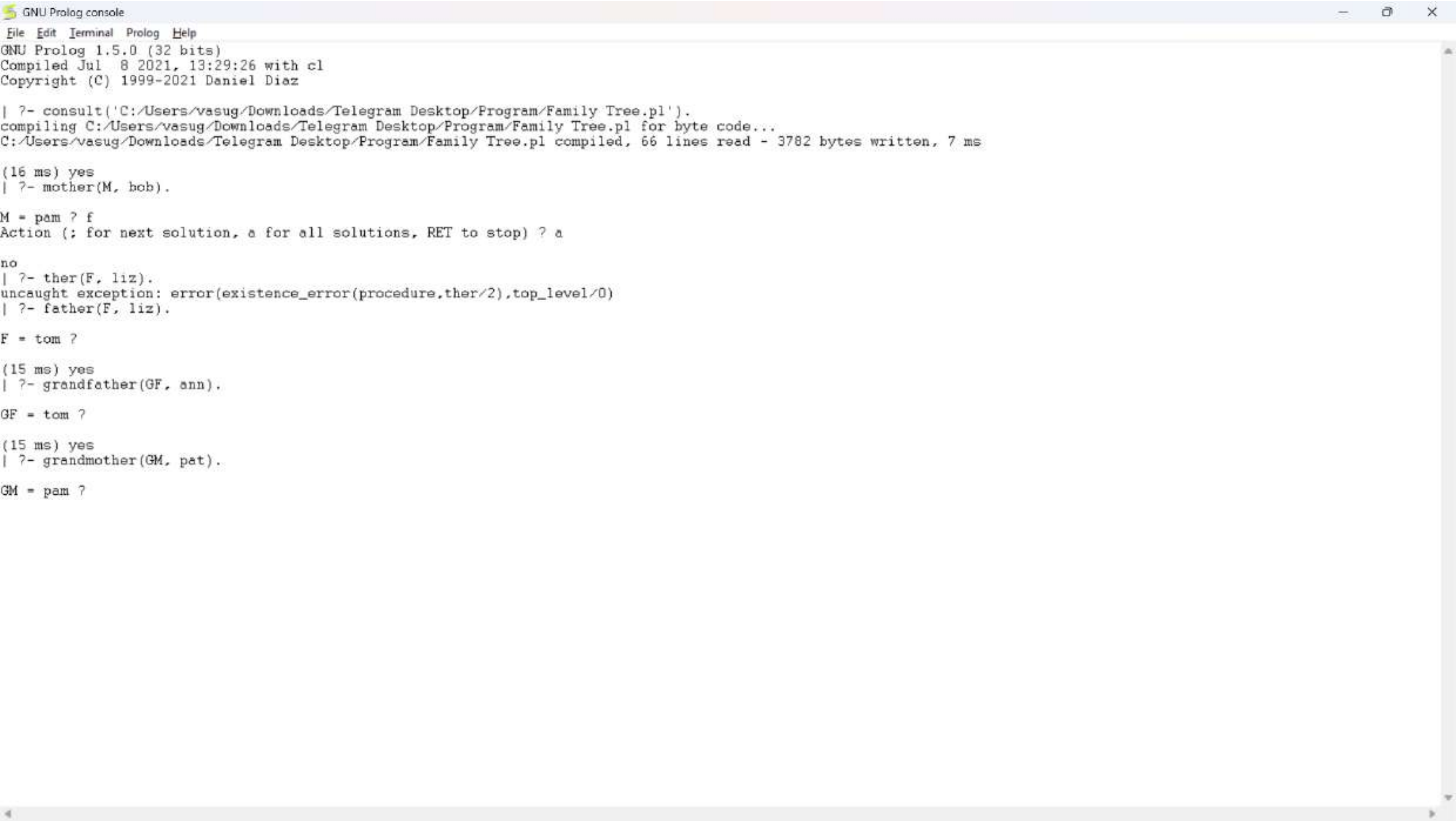
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Planet.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Planet.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Planet.pl compiled, 35 lines read - 3120 bytes written, 9 ms

yes
| ?- planet_info(mars).
Planet: mars
Distance from Sun: 227.90000000000001 million km
Radius: 3389 km
Moons: 2

yes
| ?- has_moons(P).

P = earth ?

yes
| ?- farther_than_earth(P).

P = mars ? p
Action (; for next solution, a for all solutions, RET to stop) ? l
Action (; for next solution, a for all solutions, RET to stop) ? a

P = jupiter

P = saturn

P = uranus

P = neptune

(16 ms) yes
| ?- net(Name, Distance, _, _).
```



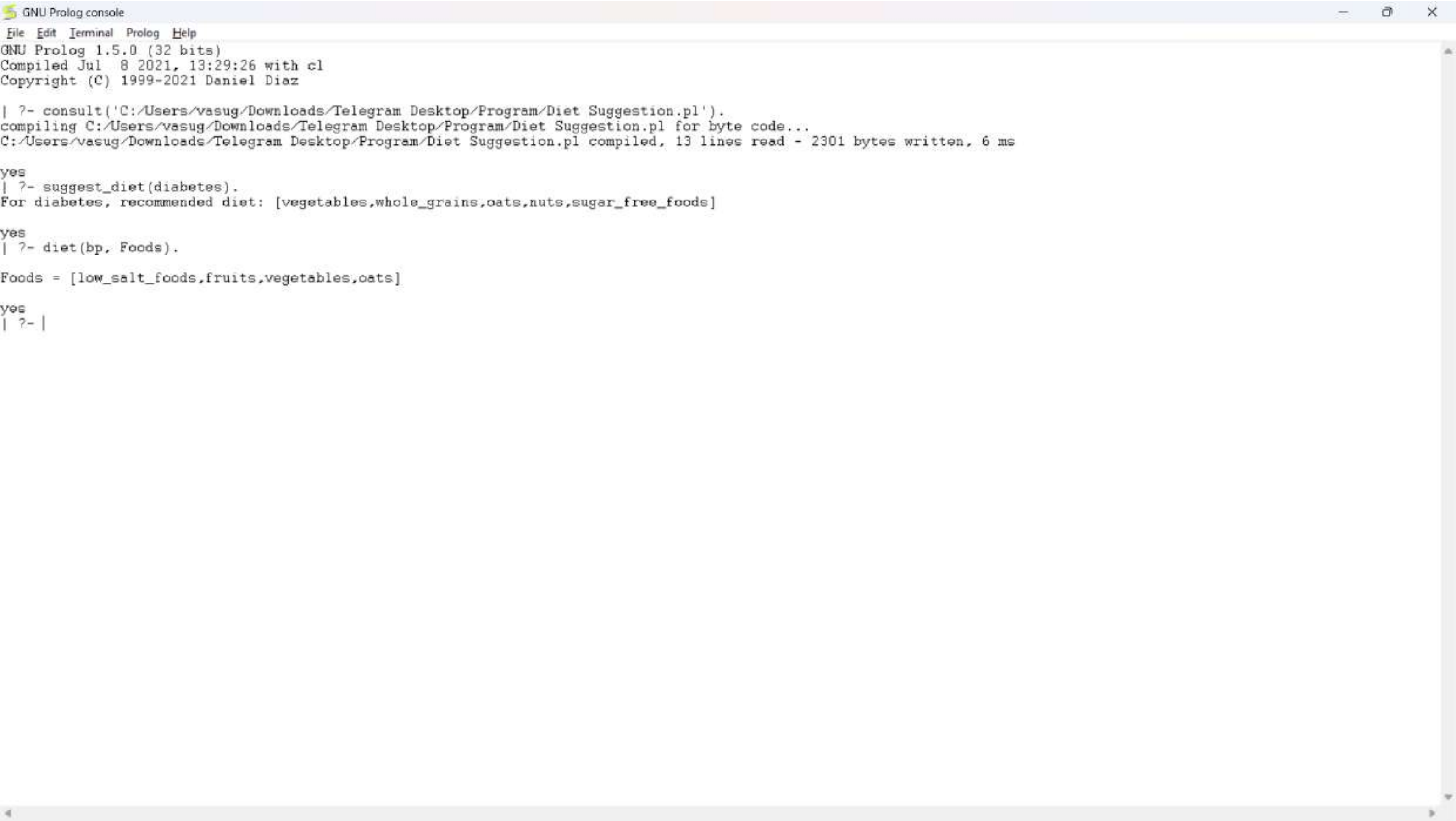
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/% hanoi(N, Source, Target, Auxiliar.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/% hanoi(N, Source, Target, Auxiliar.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/% hanoi(N, Source, Target, Auxiliar.pl compiled, 11 lines read - 1320 bytes written, 6 ms

yes
| ?- hanoi(3, left, right, middle).
Move disk 1 from left to right
Move disk 2 from left to middle
Move disk 1 from right to middle
Move disk 3 from left to right
Move disk 1 from middle to left
Move disk 2 from middle to right
Move disk 1 from left to right

(16 ms) yes
| ?- |
```



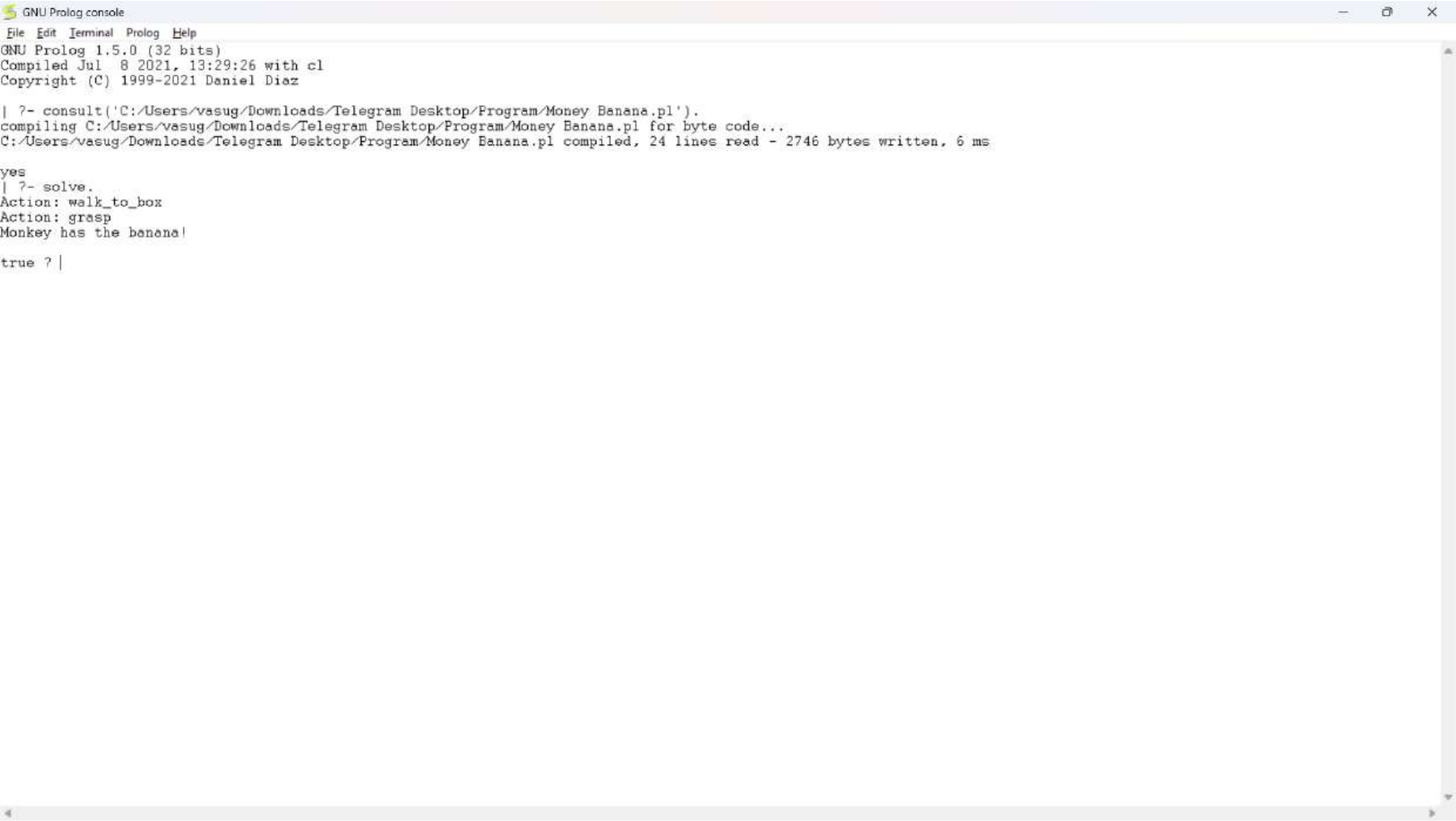
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Birds.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Birds.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Birds.pl compiled, 27 lines read - 1764 bytes written, 6 ms

(31 ms) yes
| ?- check_bird(penguin).
penguin cannot fly.

yes
| ?- check_bird(pigeon).
pigeon can fly.

true ?
```



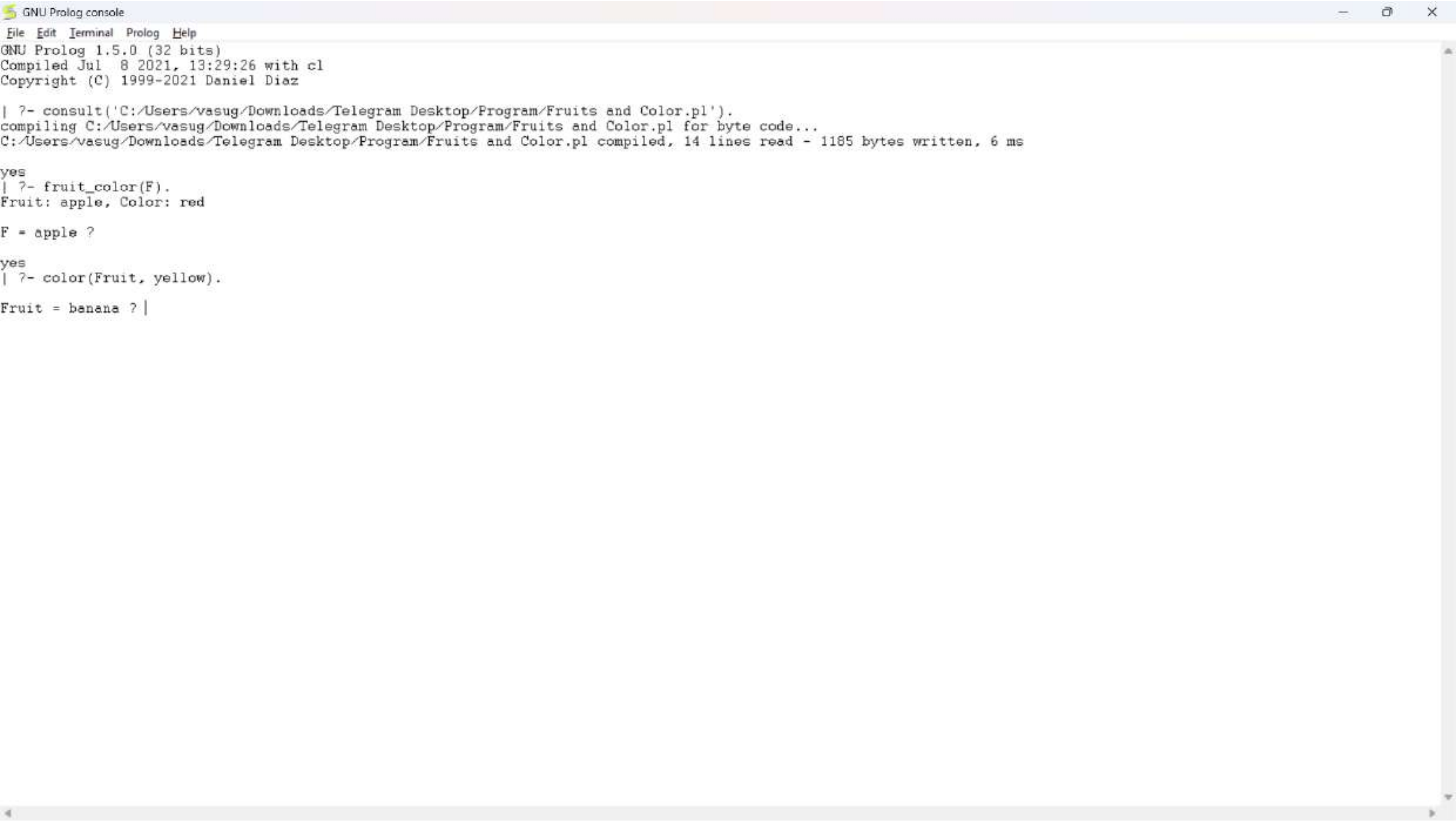
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Family Tree.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Family Tree.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Family Tree.pl compiled, 66 lines read - 3782 bytes written, 7 ms

(16 ms) yes
| ?- mother(M, bob).

M = pam ? f
Action (; for next solution, a for all solutions, RET to stop) ? a

no
| ?- ther(F, liz).
uncaught exception: error(existence_error(procedure,ther/2),top_level/0)
| ?- father(F, liz).

F = tom ?

(15 ms) yes
| ?- grandfather(GF, ann).

GF = tom ?

(15 ms) yes
| ?- grandmother(GM, pat).

GM = pam ?
```



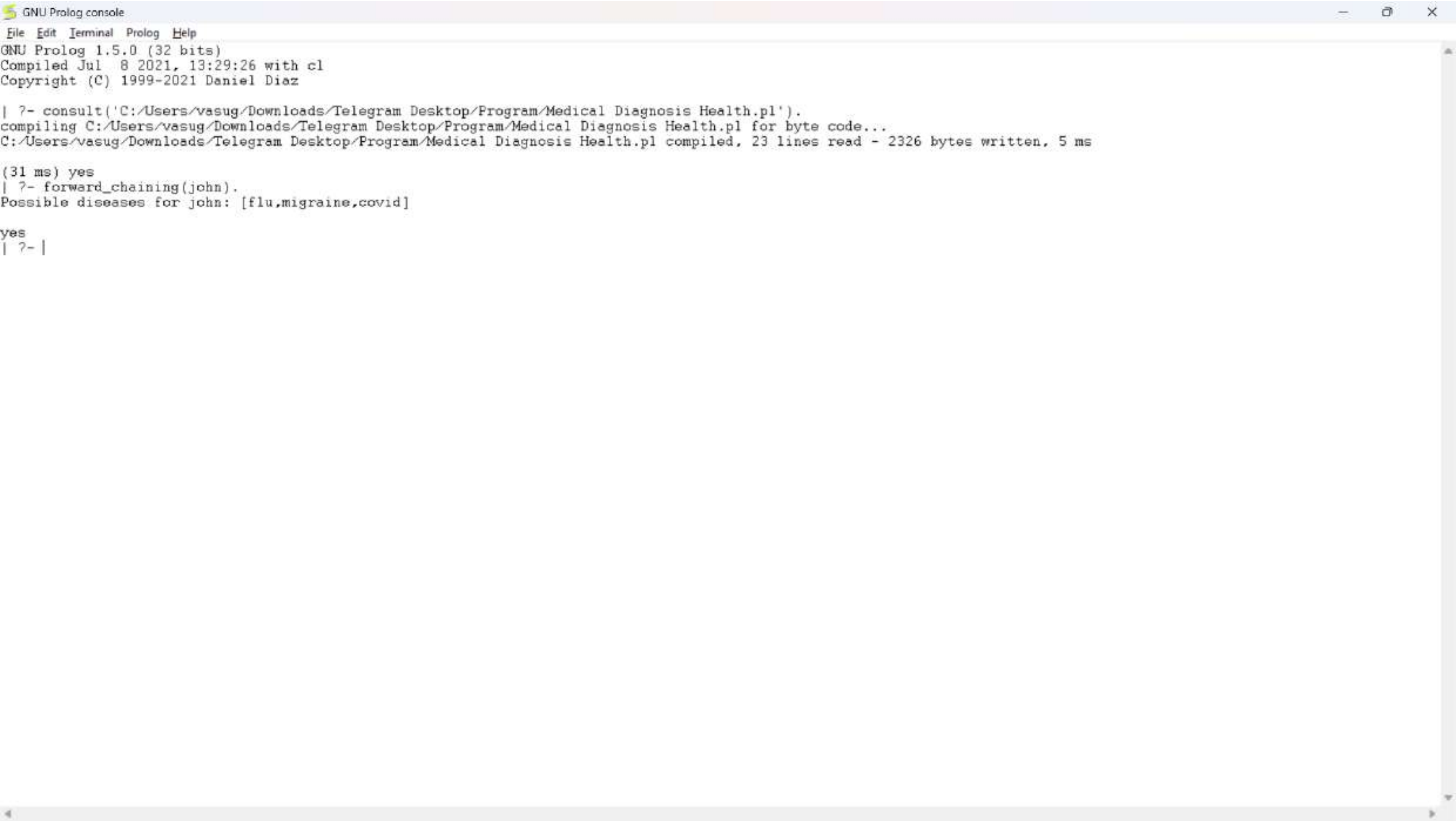
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Diet Suggestion.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Diet Suggestion.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Diet Suggestion.pl compiled, 13 lines read - 2301 bytes written, 6 ms

yes
| ?- suggest_diet(diabetes).
For diabetes, recommended diet: [vegetables,whole_grains,oats,nuts,sugar_free_foods]

yes
| ?- diet(bp, Foods).

Foods = [low_salt_foods,fruits,vegetables,oats]

yes
| ?- |
```



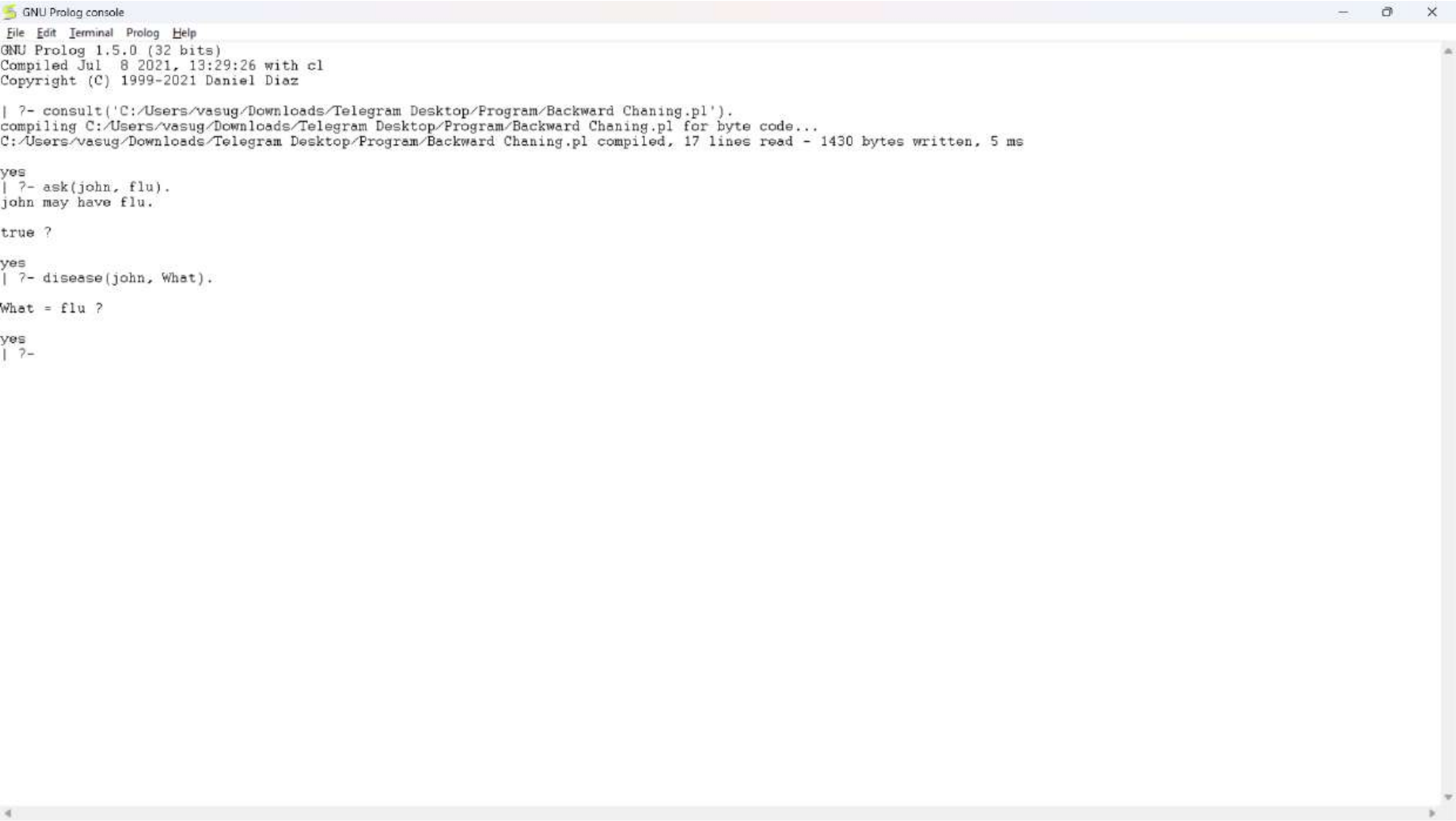
```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Money Banana.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Money Banana.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Money Banana.pl compiled, 24 lines read - 2746 bytes written, 6 ms

yes
| ?- solve.
Action: walk_to_box
Action: grasp
Monkey has the banana!

true ? |
```



```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Fruits and Color.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Fruits and Color.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Fruits and Color.pl compiled, 14 lines read - 1185 bytes written, 6 ms

yes
| ?- fruit_color(F).
Fruit: apple, Color: red

F = apple ?

yes
| ?- color(Fruit, yellow).

Fruit = banana ? 
```



```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Medical Diagnosis Health.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Medical Diagnosis Health.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Medical Diagnosis Health.pl compiled, 23 lines read - 2326 bytes written, 5 ms

(31 ms) yes
| ?- forward_chaining(john).
Possible diseases for john: [flu,migraine,covid]

yes
| ?- 
```



```
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 13:29:26 with cl
Copyright (C) 1999-2021 Daniel Diaz

| ?- consult('C:/Users/vasug/Downloads/Telegram Desktop/Program/Backward Chaning.pl').
compiling C:/Users/vasug/Downloads/Telegram Desktop/Program/Backward Chaning.pl for byte code...
C:/Users/vasug/Downloads/Telegram Desktop/Program/Backward Chaning.pl compiled, 17 lines read - 1430 bytes written, 5 ms

yes
| ?- ask(john, flu).
john may have flu.

true ?

yes
| ?- disease(john, What).

What = flu ?

yes
| ?-
```