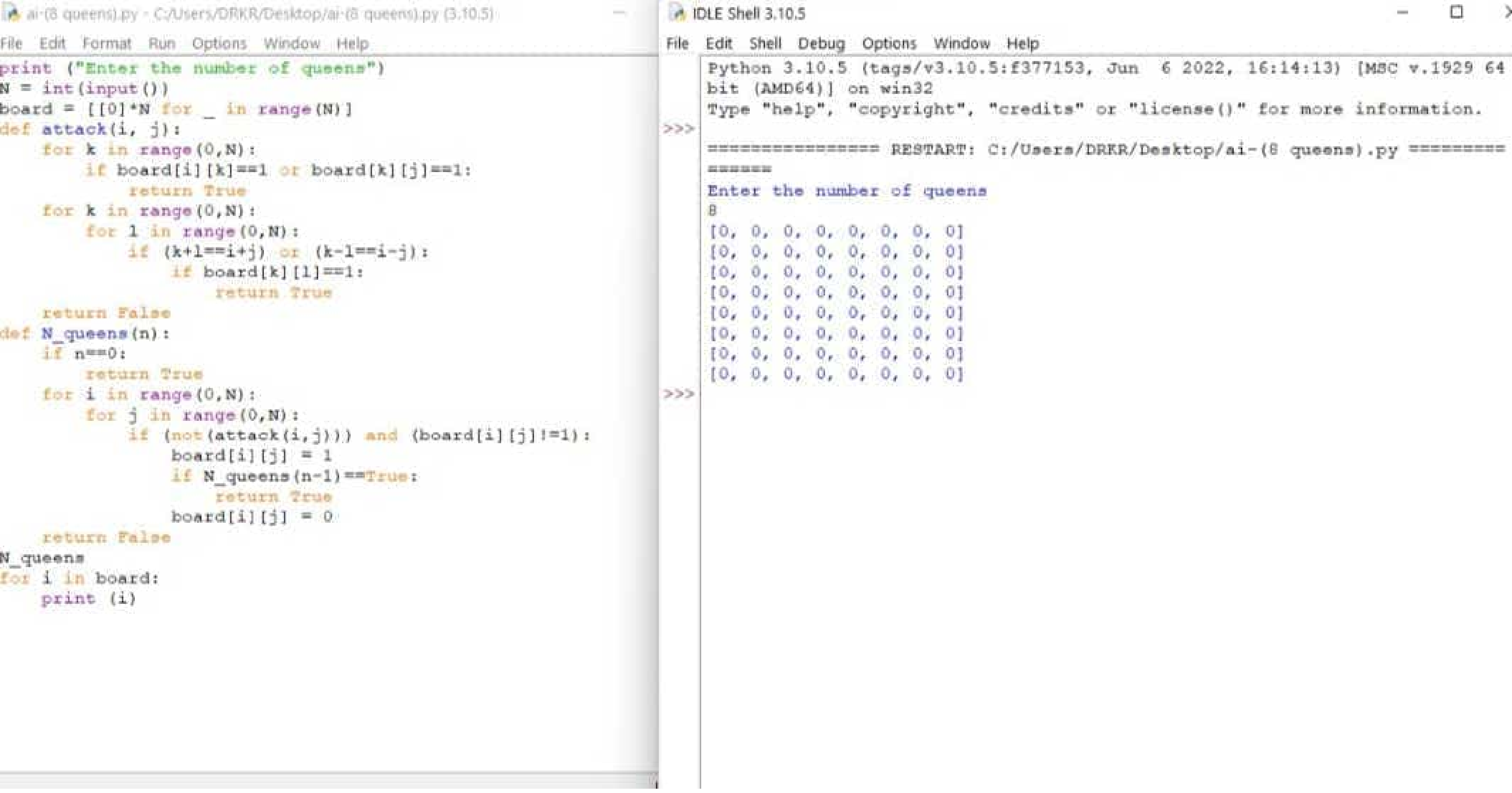
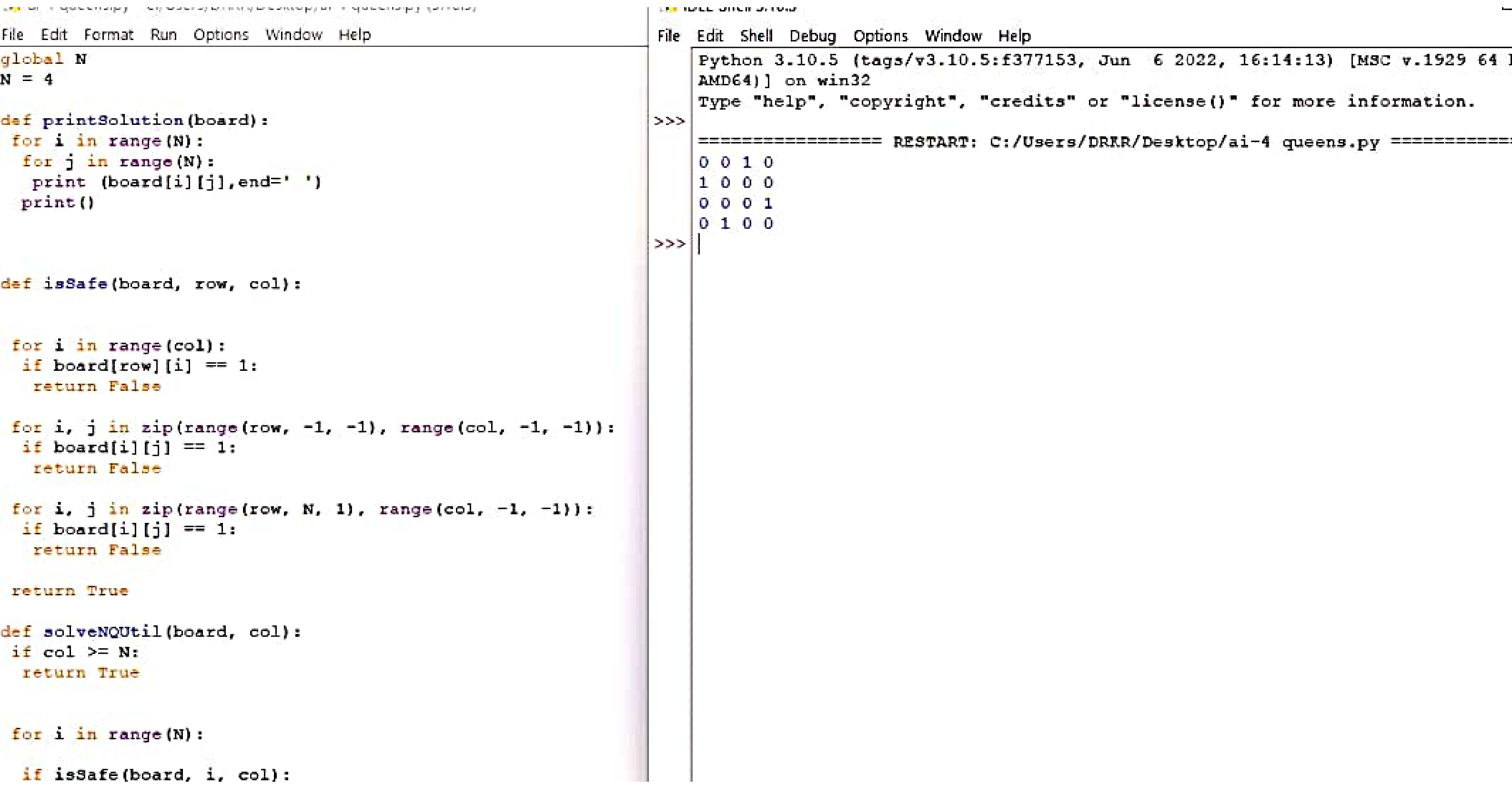

ai-(8 queens).py - C:/Users/DRKR/Desktop/ai-(8 queens).py (3.10.5)
File Edit Format Run Options Window Help
IDLE Shell 3.10.5
File Edit Shell Debug Options Window Help

```python
print ("Enter the number of queens")
N = int(input())
board = [[0]*N for _ in range(N)]
def attack(i, j):
    for k in range(0,N):
        if board[i][k]==1 or board[k][j]==1:
            return True
    for k in range(0,N):
        for l in range(0,N):
            if (k+l==i+j) or (k-l==i-j):
                if board[k][l]==1:
                    return True
    return False
def N_queens(n):
    if n==0:
        return True
    for i in range(0,N):
        for j in range(0,N):
            if (not(attack(i,j))) and (board[i][j]!=1):
                board[i][j] = 1
                if N_queens(n-1)==True:
                    return True
                board[i][j] = 0
    return False
N_queens
for i in board:
    print (i)
```

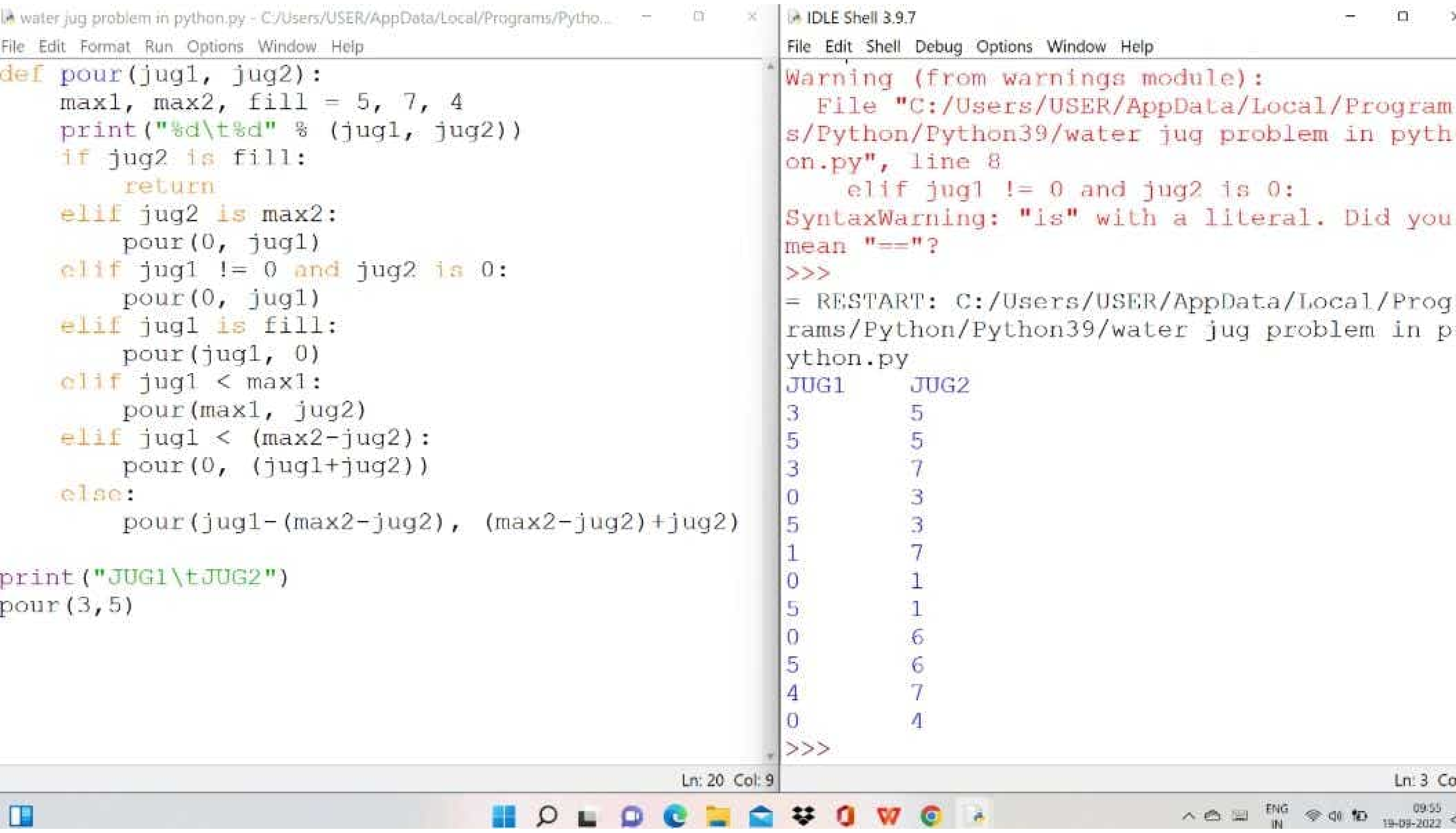
```
    Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64
    bit (AMD64)] on win32
    Type "help", "copyright", "credits" or "license()" for more information.
>>>
    ================ RESTART: C:/Users/DRKR/Desktop/ai-(8 queens).py =========
    ======
    Enter the number of queens
    8
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
    [0, 0, 0, 0, 0, 0, 0, 0]
>>>
```



File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help

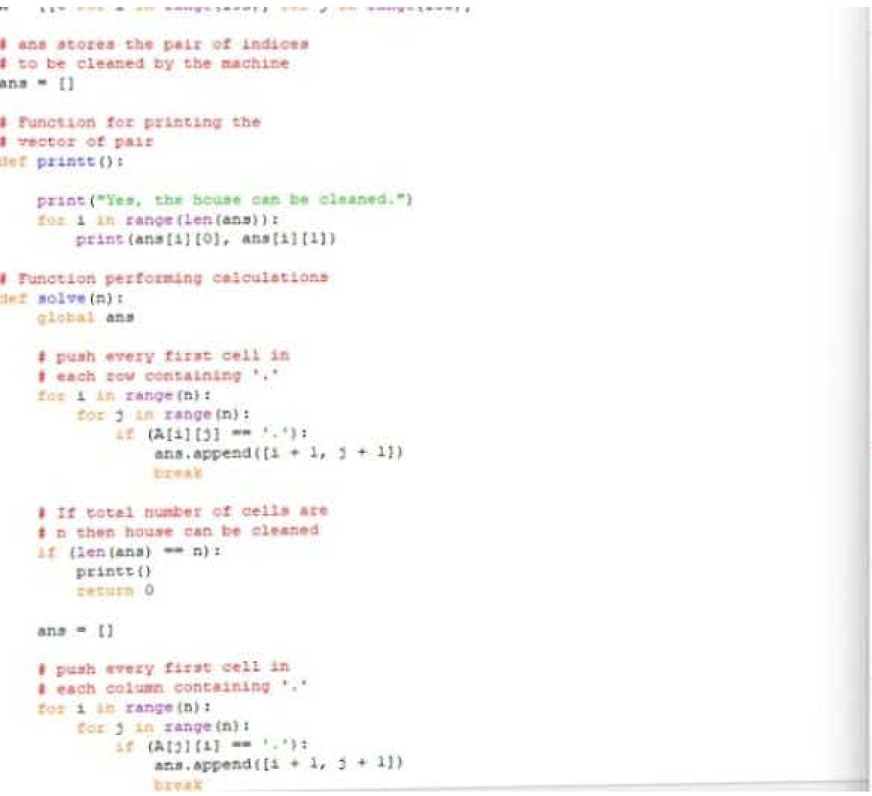
```
global N
N = 4

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):
```

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: C:/Users/DRRR/Desktop/ai-4 queens.py ===========
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
>>>
```

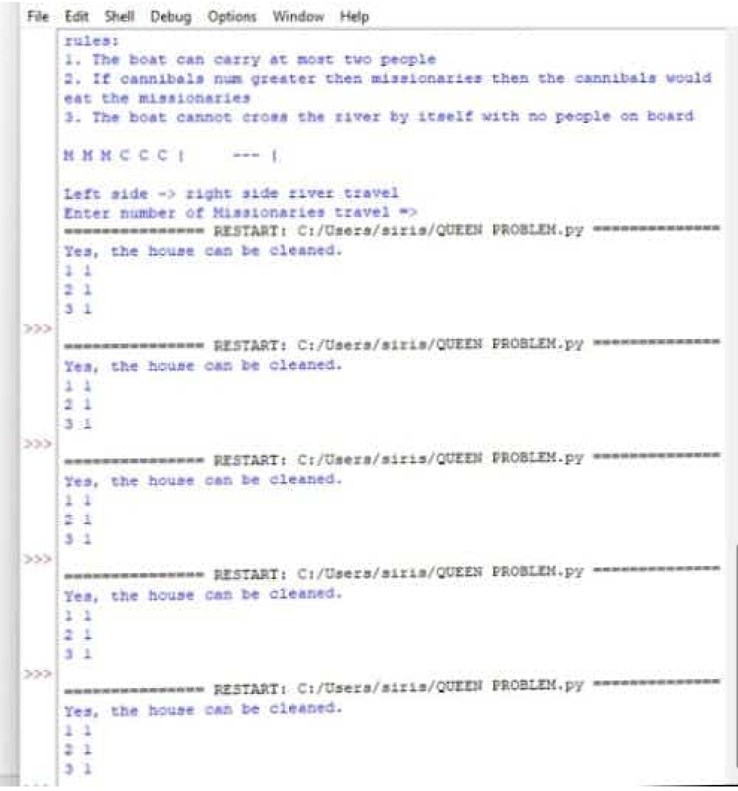

water jug problem in python.py - C:/Users/USER/AppData/Local/Programs/Pytho...
File Edit Format Run Options Window Help
IDLE Shell 3.9.7
File Edit Shell Debug Options Window Help
Ln: 20 Col: 9
Ln: 3 Co
ENG IN
09:55 19-09-2022

```
def pour(jug1, jug2):
    max1, max2, fill = 5, 7, 4
    print("%d\t%d" % (jug1, jug2))
    if jug2 is fill:
        return
    elif jug2 is max2:
        pour(0, jug1)
    elif jug1 != 0 and jug2 is 0:
        pour(0, jug1)
    elif jug1 is fill:
        pour(jug1, 0)
    elif jug1 < max1:
        pour(max1, jug2)
    elif jug1 < (max2-jug2):
        pour(0, (jug1+jug2))
    else:
        pour(jug1-(max2-jug2), (max2-jug2)+jug2)

print("JUG1\tJUG2")
pour(3,5)
```

```
Warning (from warnings module):
  File "C:/Users/USER/AppData/Local/Program
s/Python/Python39/water jug problem in pyth
on.py", line 8
    elif jug1 != 0 and jug2 is 0:
SyntaxWarning: "is" with a literal. Did you
mean "=="?
>>>
= RESTART: C:/Users/USER/AppData/Local/Prog
rams/Python/Python39/water jug problem in p
ython.py
JUG1    JUG2
3       5
5       5
3       7
0       3
5       3
1       7
0       1
5       1
0       6
5       6
4       7
0       4
>>>
```



```
A = [[0 for i in range(100)] for j in range(100)]

# ans stores the pair of indices
# to be cleaned by the machine
ans = []

# Function for printing the
# vector of pair
def printt():

    print("Yes, the house can be cleaned.")
    for i in range(len(ans)):
        print(ans[i][0], ans[i][1])

# Function performing calculations
def solve(n):
    global ans

    # push every first cell in
    # each row containing '.'
    for i in range(n):
        for j in range(n):
            if (A[i][j] == '.'):
                ans.append([i + 1, j + 1])
                break

    # If total number of cells are
    # n then house can be cleaned
    if (len(ans) == n):
        printt()
        return 0

    ans = []

    # push every first cell in
    # each column containing '.'
    for i in range(n):
        for j in range(n):
            if (A[j][i] == '.'):
                ans.append([i + 1, j + 1])
                break
```



```
rules:
1. The boat can carry at most two people
2. If cannibals num greater then missionaries then the cannibals would
eat the missionaries
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 =>
=============== RESTART: C:/Users/siris/QUEEN PROBLEM.py ==============
Yes, the house can be cleaned.
1 1
2 1
3 1
>>>
=============== RESTART: C:/Users/siris/QUEEN PROBLEM.py ==============
Yes, the house can be cleaned.
1 1
2 1
3 1
>>>
=============== RESTART: C:/Users/siris/QUEEN PROBLEM.py ==============
Yes, the house can be cleaned.
1 1
2 1
3 1
>>>
=============== RESTART: C:/Users/siris/QUEEN PROBLEM.py ==============
Yes, the house can be cleaned.
1 1
2 1
3 1
>>>
=============== RESTART: C:/Users/siris/QUEEN PROBLEM.py ==============
Yes, the house can be cleaned.
1 1
2 1
3 1
```