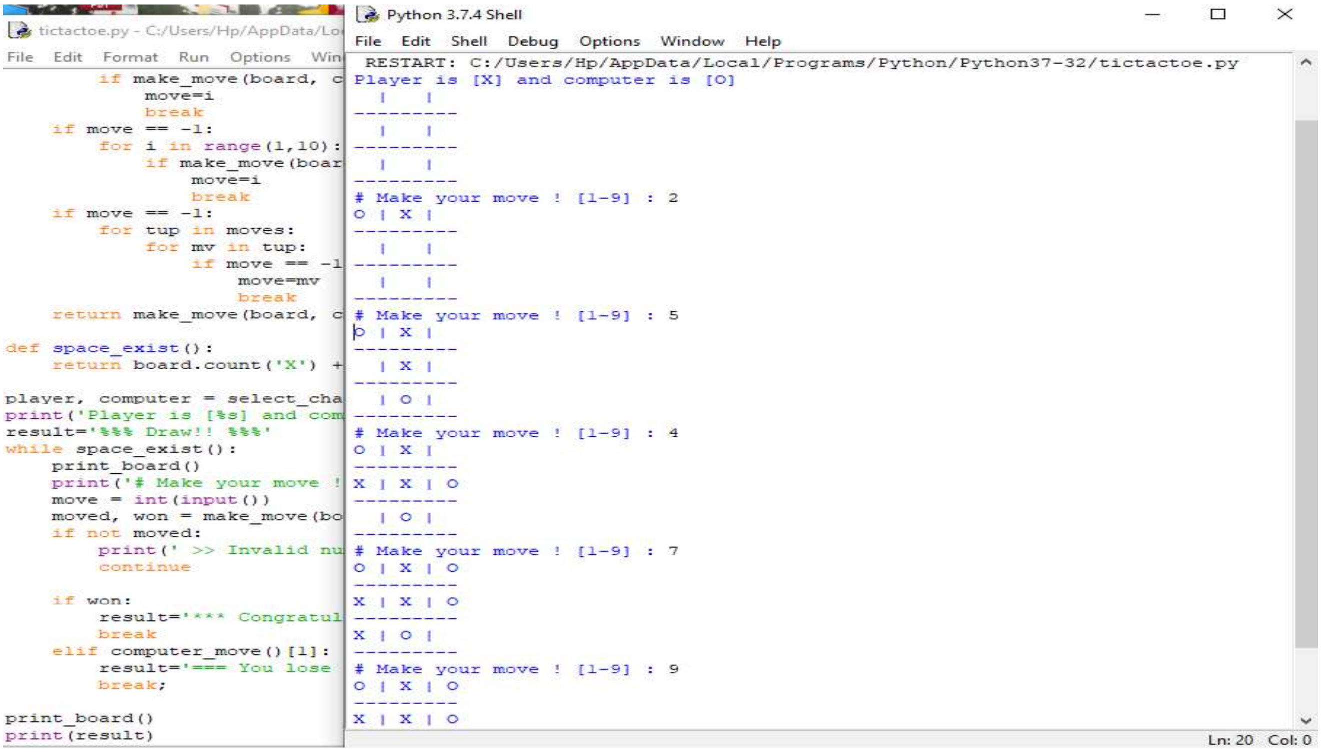
Salina M S Mammy

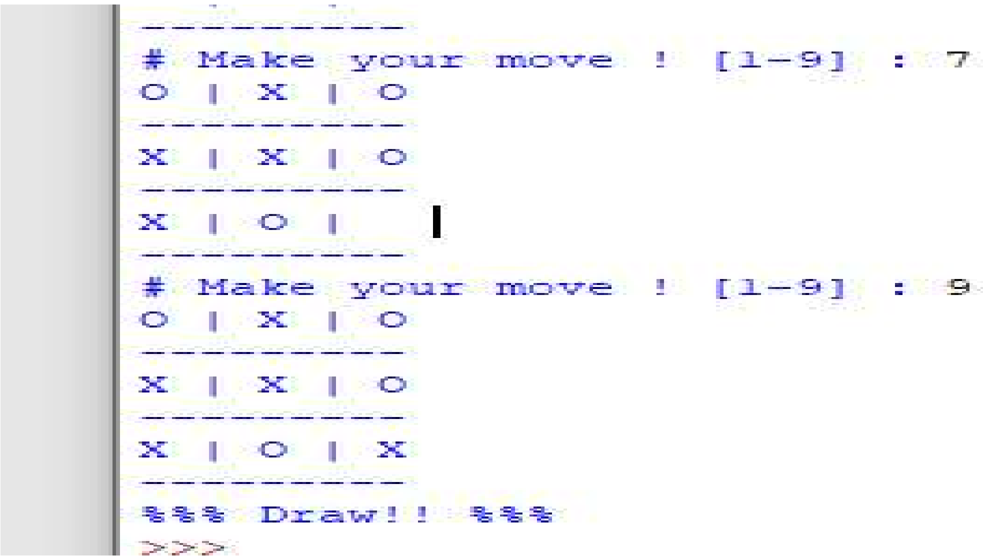
1BM18CS417

1. Demonstrate the program Tic-Tac-Toe game.


tictactoe.py - C:/Users/Hp/AppData/Lo
File Edit Format Run Options Win
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Ln: 20 Col: 0

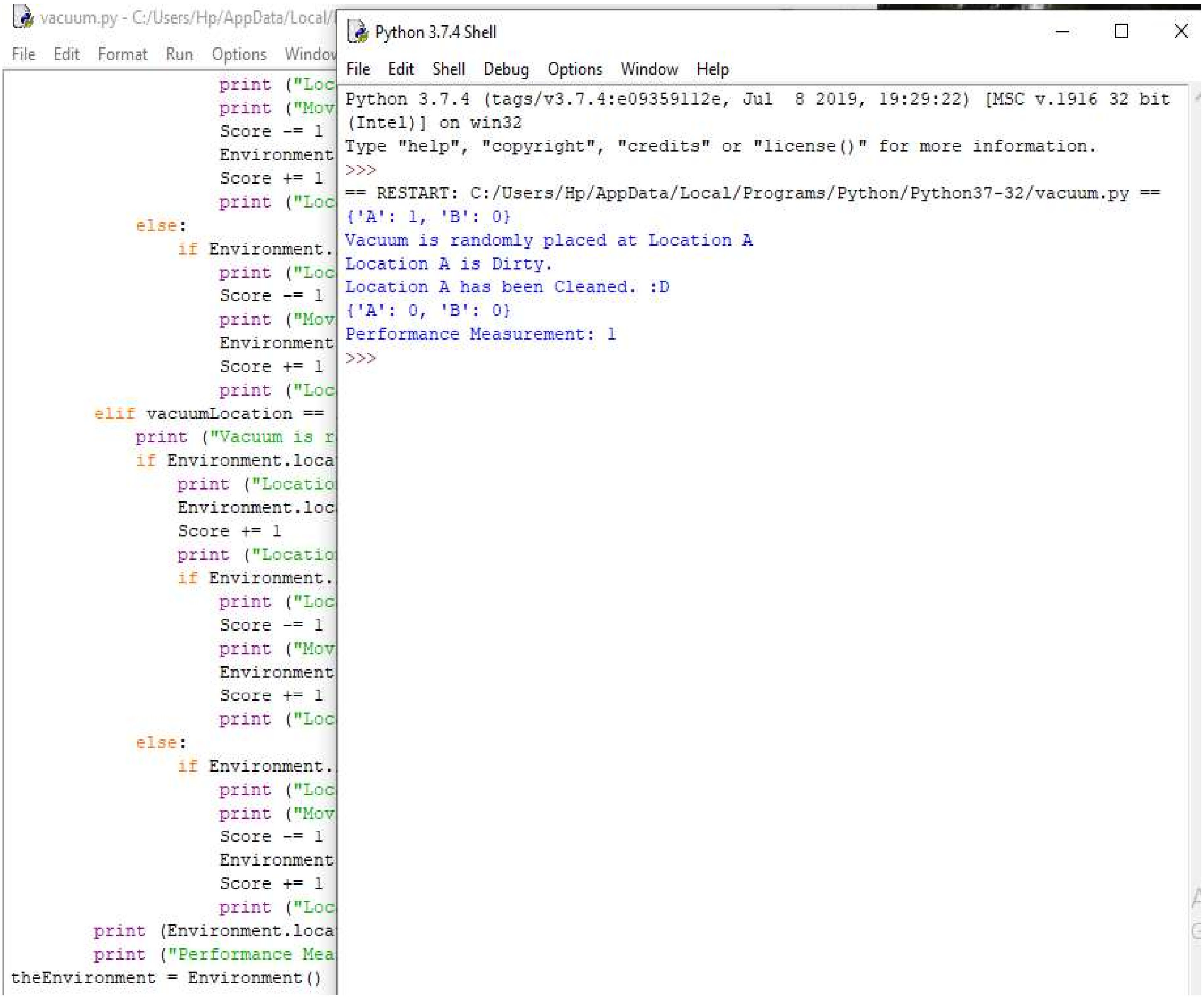
```
            if make_move(board, c
                move=i
                break
        if move == -1:
            for i in range(1,10):
                if make_move(boar
                    move=i
                    break
        if move == -1:
            for tup in moves:
                for mv in tup:
                    if move == -1
                        move=mv
                        break
        return make_move(board, c

def space_exist():
        return board.count('X') +

player, computer = select_cha
print('Player is [%s] and com
result='%%% Draw!! %%%'
while space_exist():
        print_board()
        print('# Make your move !
        move = int(input())
        moved, won = make_move(bo
        if not moved:
            print(' >> Invalid nu
            continue

        if won:
            result='*** Congratul
            break
        elif computer_move()[1]:
            result='=== You lose
            break;

print_board()
print(result)
```

```
 RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/tictactoe.py
Player is [X] and computer is [O]
  |   |
---------
  |   |
---------
  |   |
---------
# Make your move ! [1-9] : 2
O | X |
---------
  |   |
---------
  |   |
---------
# Make your move ! [1-9] : 5
O | X |
---------
  | X |
---------
  | O |
---------
# Make your move ! [1-9] : 4
O | X |
---------
X | X | O
---------
  | O |
---------
# Make your move ! [1-9] : 7
O | X | O
---------
X | X | O
---------
X | O |
---------
# Make your move ! [1-9] : 9
O | X | O
---------
X | X | O
```



```
---------
# Make your move ! [1-9] : 7
O | X | O
---------
X | X | O
---------
X | O |
---------
# Make your move ! [1-9] : 9
O | X | O
---------
X | X | O
---------
X | O | X
---------
%%% Draw!! %%%
>>>
```

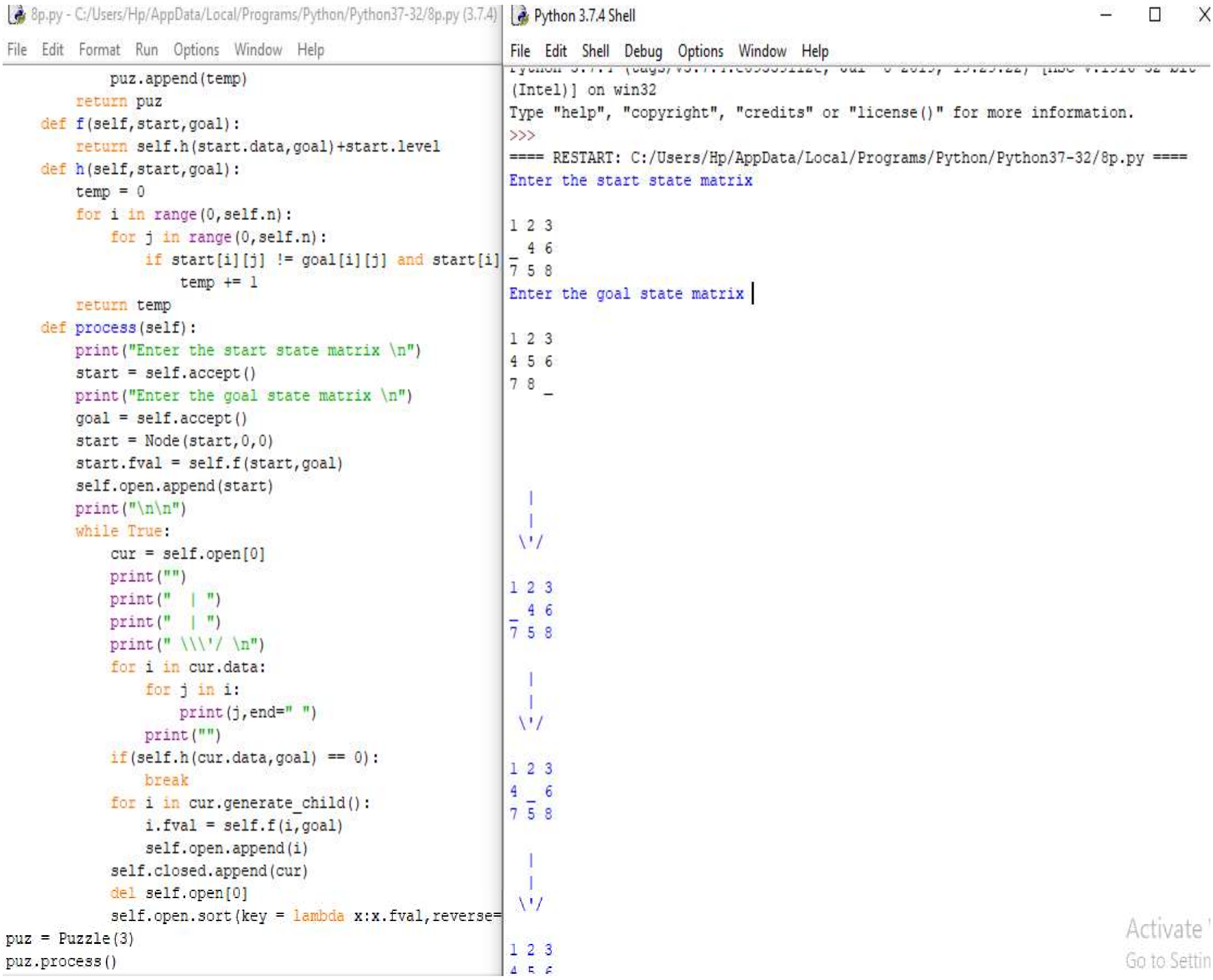
2. Solve Vacuum cleaner agent


vacuum.py - C:/Users/Hp/AppData/Local/
File Edit Format Run Options Window
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

```
                    print ("Loc
                    print ("Mov
                    Score -= 1
                    Environment
                    Score += 1
                    print ("Loc
            else:
                if Environment.
                    print ("Loc
                    Score -= 1
                    print ("Mov
                    Environment
                    Score += 1
                    print ("Loc
        elif vacuumLocation ==
            print ("Vacuum is r
            if Environment.loca
                print ("Locatio
                Environment.loc
                Score += 1
                print ("Locatio
                if Environment.
                    print ("Loc
                    Score -= 1
                    print ("Mov
                    Environment
                    Score += 1
                    print ("Loc
            else:
                if Environment.
                    print ("Loc
                    print ("Mov
                    Score -= 1
                    Environment
                    Score += 1
                    print ("Loc
        print (Environment.loca
        print ("Performance Mea
theEnvironment = Environment()
```

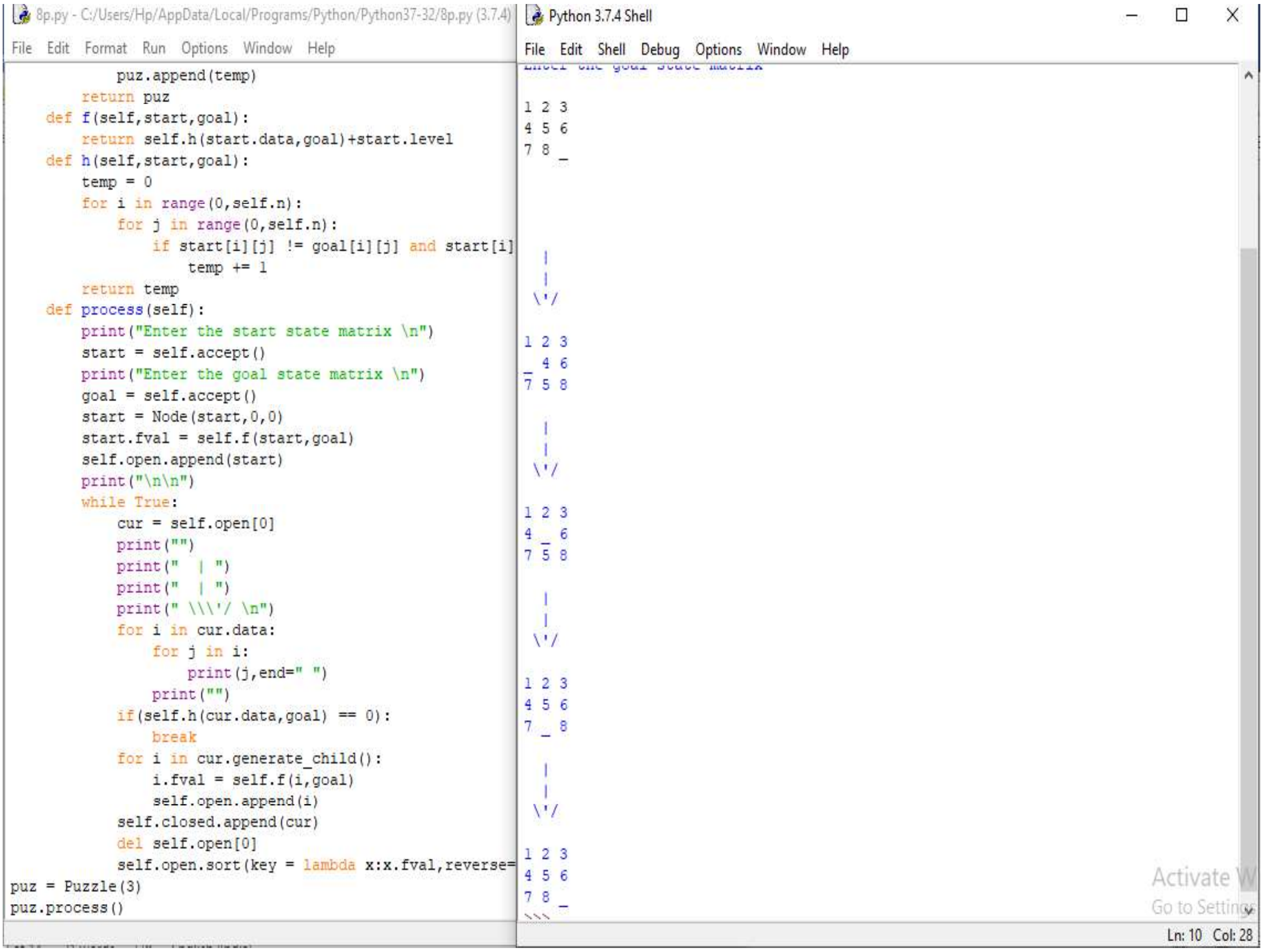
```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/vacuum.py ==
{'A': 1, 'B': 0}
Vacuum is randomly placed at Location A
Location A is Dirty.
Location A has been Cleaned. :D
{'A': 0, 'B': 0}
Performance Measurement: 1
>>> 
```

3. Solve 8 puzzle problem


8p.py - C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/8p.py (3.7.4)
File Edit Format Run Options Window Help
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

```
            puz.append(temp)
        return puz
    def f(self,start,goal):
        return self.h(start.data,goal)+start.level
    def h(self,start,goal):
        temp = 0
        for i in range(0,self.n):
            for j in range(0,self.n):
                if start[i][j] != goal[i][j] and start[i]
                    temp += 1
        return temp
    def process(self):
        print("Enter the start state matrix \n")
        start = self.accept()
        print("Enter the goal state matrix \n")
        goal = self.accept()
        start = Node(start,0,0)
        start.fval = self.f(start,goal)
        self.open.append(start)
        print("\n\n")
        while True:
            cur = self.open[0]
            print("")
            print("  | ")
            print("  | ")
            print(" \\\'/ \n")
            for i in cur.data:
                for j in i:
                    print(j,end=" ")
                print("")
            if(self.h(cur.data,goal) == 0):
                break
            for i in cur.generate_child():
                i.fval = self.f(i,goal)
                self.open.append(i)
            self.closed.append(cur)
            del self.open[0]
            self.open.sort(key = lambda x:x.fval,reverse=
puz = Puzzle(3)
puz.process()
```

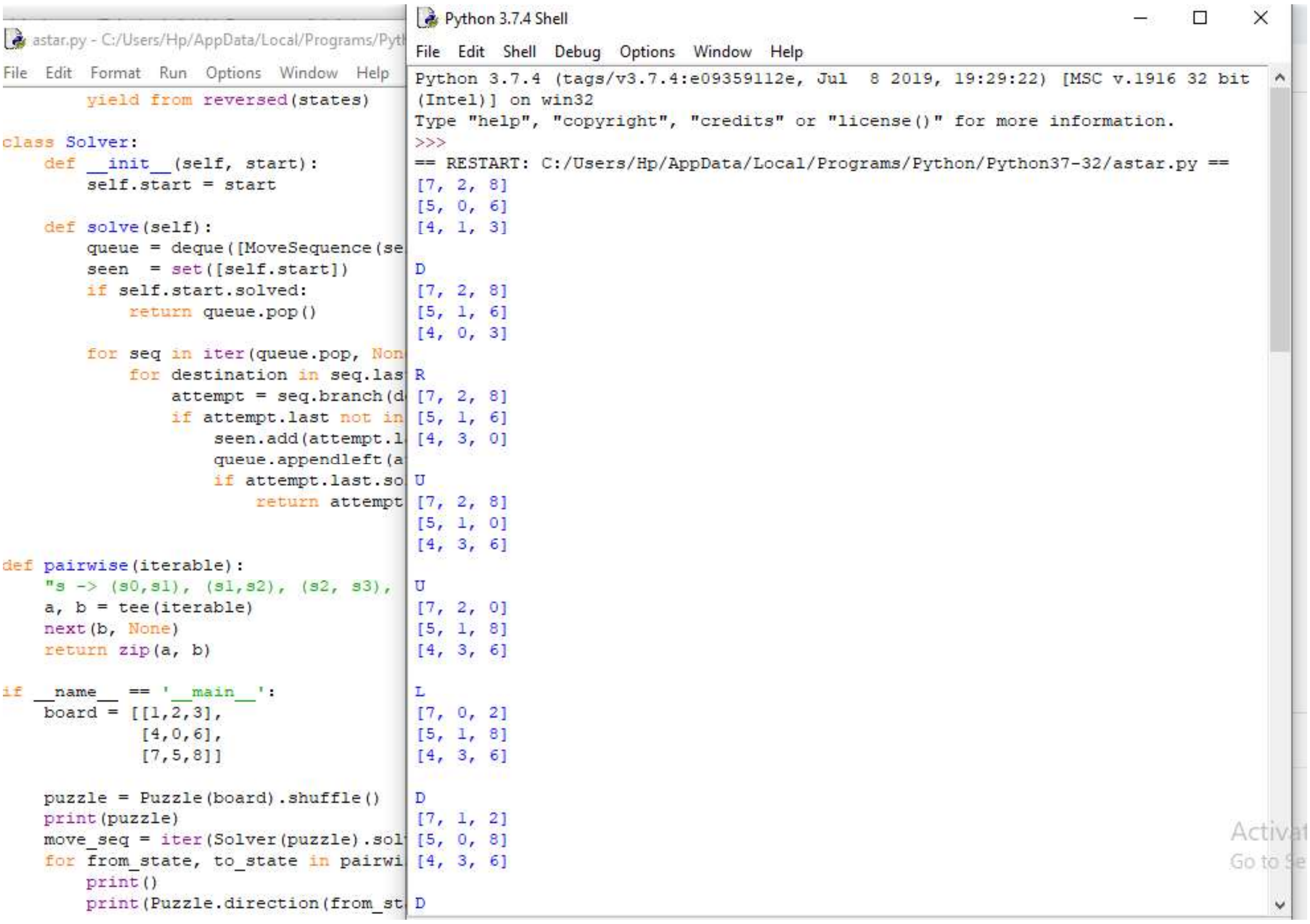
```
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/8p.py ====
Enter the start state matrix

1 2 3
_ 4 6
7 5 8
Enter the goal state matrix

1 2 3
4 5 6
7 8 _




  |
  |
 \'/

1 2 3
_ 4 6
7 5 8

  |
  |
 \'/

1 2 3
4 _ 6
7 5 8

  |
  |
 \'/

1 2 3
4 5 6
```


8p.py - C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/8p.py (3.7.4)
File Edit Format Run Options Window Help
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Activate W
Go to Setting
Ln: 10 Col: 28

```
            puz.append(temp)
        return puz
    def f(self,start,goal):
        return self.h(start.data,goal)+start.level
    def h(self,start,goal):
        temp = 0
        for i in range(0,self.n):
            for j in range(0,self.n):
                if start[i][j] != goal[i][j] and start[i]
                    temp += 1
        return temp
    def process(self):
        print("Enter the start state matrix \n")
        start = self.accept()
        print("Enter the goal state matrix \n")
        goal = self.accept()
        start = Node(start,0,0)
        start.fval = self.f(start,goal)
        self.open.append(start)
        print("\n\n")
        while True:
            cur = self.open[0]
            print("")
            print("  | ")
            print("  | ")
            print(" \\\'/ \n")
            for i in cur.data:
                for j in i:
                    print(j,end=" ")
                print("")
            if(self.h(cur.data,goal) == 0):
                break
            for i in cur.generate_child():
                i.fval = self.f(i,goal)
                self.open.append(i)
            self.closed.append(cur)
            del self.open[0]
            self.open.sort(key = lambda x:x.fval,reverse=
puz = Puzzle(3)
puz.process()
```

```
Enter the goal state matrix

1 2 3
4 5 6
7 8 _



  |
  |
 \'/

1 2 3
_ 4 6
7 5 8

  |
  |
 \'/

1 2 3
4 _ 6
7 5 8

  |
  |
 \'/

1 2 3
4 5 6
7 _ 8

  |
  |
 \'/

1 2 3
4 5 6
7 8 _
>>>
```

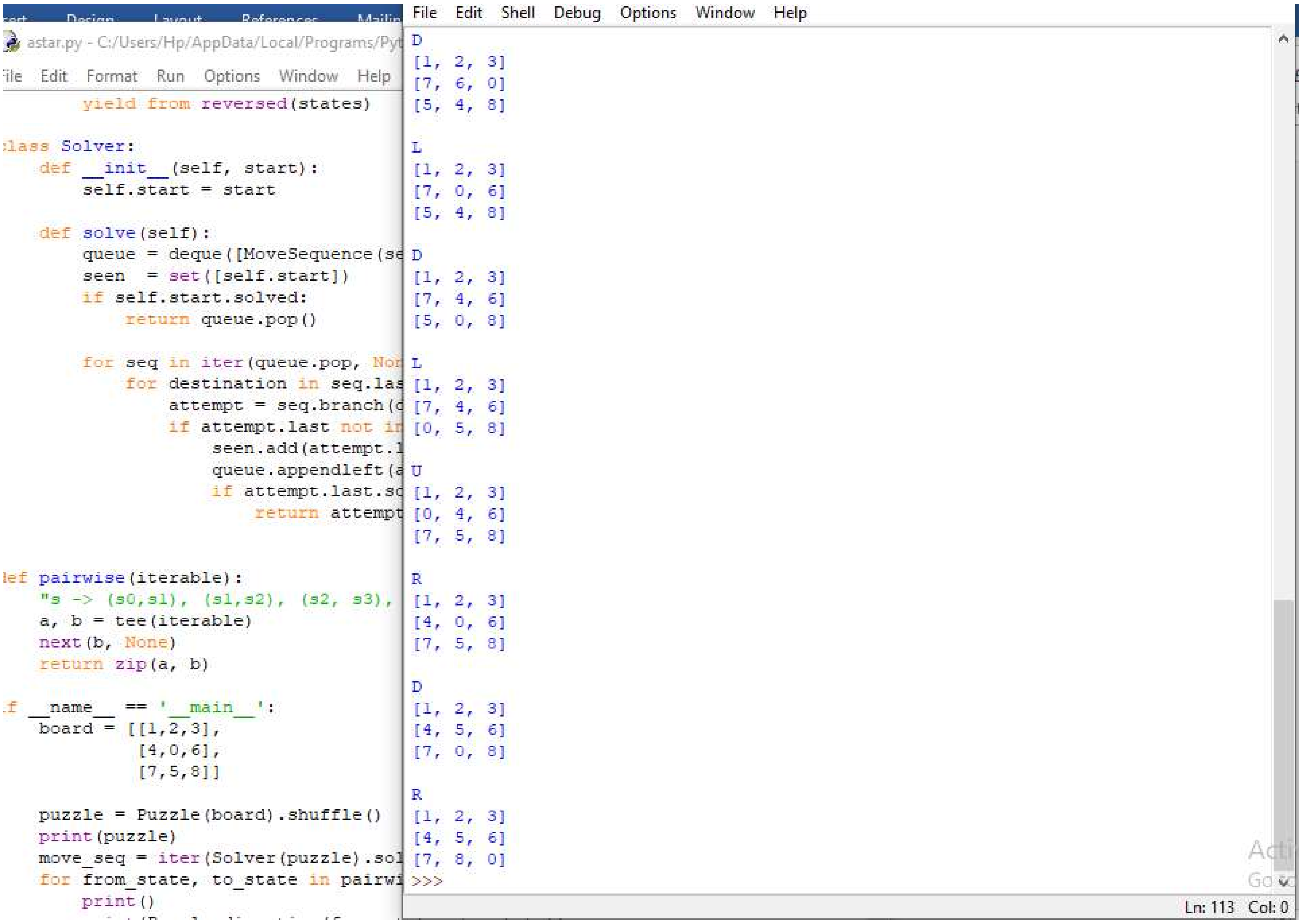
4. Implement A* Search



astar.py - C:/Users/Hp/AppData/Local/Programs/Pyth
File Edit Format Run Options Window Help
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

```
        yield from reversed(states)

class Solver:
    def __init__(self, start):
        self.start = start

    def solve(self):
        queue = deque([MoveSequence(se
        seen  = set([self.start])
        if self.start.solved:
            return queue.pop()

        for seq in iter(queue.pop, Non
            for destination in seq.las
                attempt = seq.branch(d
                if attempt.last not in
                    seen.add(attempt.l
                    queue.appendleft(a
                    if attempt.last.so
                        return attempt


def pairwise(iterable):
    "s -> (s0,s1), (s1,s2), (s2, s3),
    a, b = tee(iterable)
    next(b, None)
    return zip(a, b)

if __name__ == '__main__':
    board = [[1,2,3],
             [4,0,6],
             [7,5,8]]

    puzzle = Puzzle(board).shuffle()
    print(puzzle)
    move_seq = iter(Solver(puzzle).sol
    for from_state, to_state in pairwi
        print()
        print(Puzzle.direction(from_st
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/astar.py ==
[7, 2, 8]
[5, 0, 6]
[4, 1, 3]

D
[7, 2, 8]
[5, 1, 6]
[4, 0, 3]

R
[7, 2, 8]
[5, 1, 6]
[4, 3, 0]

U
[7, 2, 8]
[5, 1, 0]
[4, 3, 6]

U
[7, 2, 0]
[5, 1, 8]
[4, 3, 6]

L
[7, 0, 2]
[5, 1, 8]
[4, 3, 6]

D
[7, 1, 2]
[5, 0, 8]
[4, 3, 6]

D
```

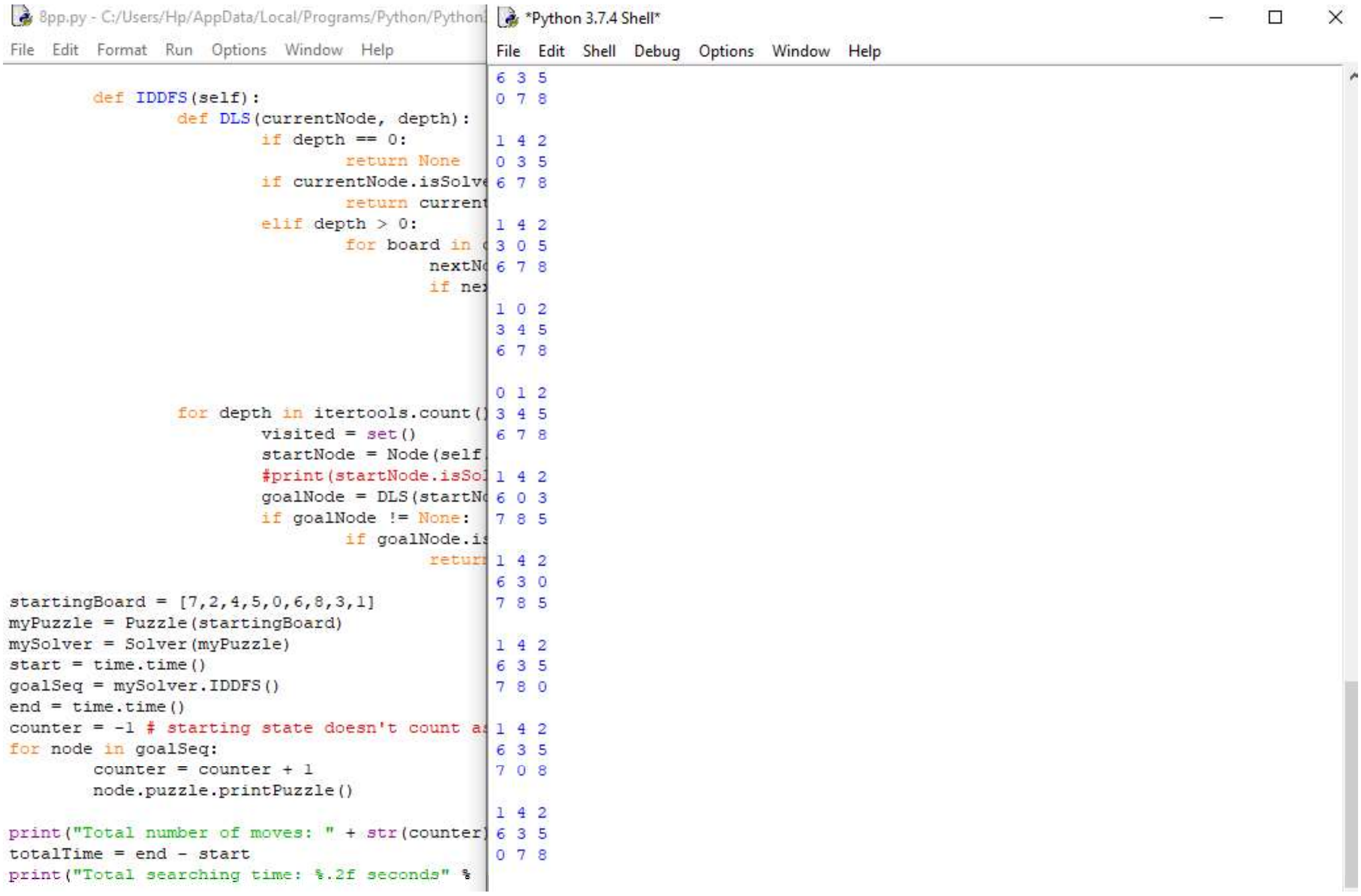


astar.py - C:/Users/Hp/AppData/Local/Programs/Pyt
File Edit Format Run Options Window Help
File Edit Shell Debug Options Window Help
Ln: 113 Col: 0

```
        yield from reversed(states)

class Solver:
    def __init__(self, start):
        self.start = start

    def solve(self):
        queue = deque([MoveSequence(se
        seen  = set([self.start])
        if self.start.solved:
            return queue.pop()

        for seq in iter(queue.pop, Nor
            for destination in seq.las
                attempt = seq.branch(d
                if attempt.last not in
                    seen.add(attempt.l
                    queue.appendleft(a
                    if attempt.last.so
                        return attempt


def pairwise(iterable):
    "s -> (s0,s1), (s1,s2), (s2, s3),
    a, b = tee(iterable)
    next(b, None)
    return zip(a, b)

if __name__ == '__main__':
    board = [[1,2,3],
             [4,0,6],
             [7,5,8]]

    puzzle = Puzzle(board).shuffle()
    print(puzzle)
    move_seq = iter(Solver(puzzle).sol
    for from_state, to_state in pairwi
        print()
```

```
D
[1, 2, 3]
[7, 6, 0]
[5, 4, 8]

L
[1, 2, 3]
[7, 0, 6]
[5, 4, 8]

D
[1, 2, 3]
[7, 4, 6]
[5, 0, 8]

L
[1, 2, 3]
[7, 4, 6]
[0, 5, 8]

U
[1, 2, 3]
[0, 4, 6]
[7, 5, 8]

R
[1, 2, 3]
[4, 0, 6]
[7, 5, 8]

D
[1, 2, 3]
[4, 5, 6]
[7, 0, 8]

R
[1, 2, 3]
[4, 5, 6]
[7, 8, 0]
>>>
```

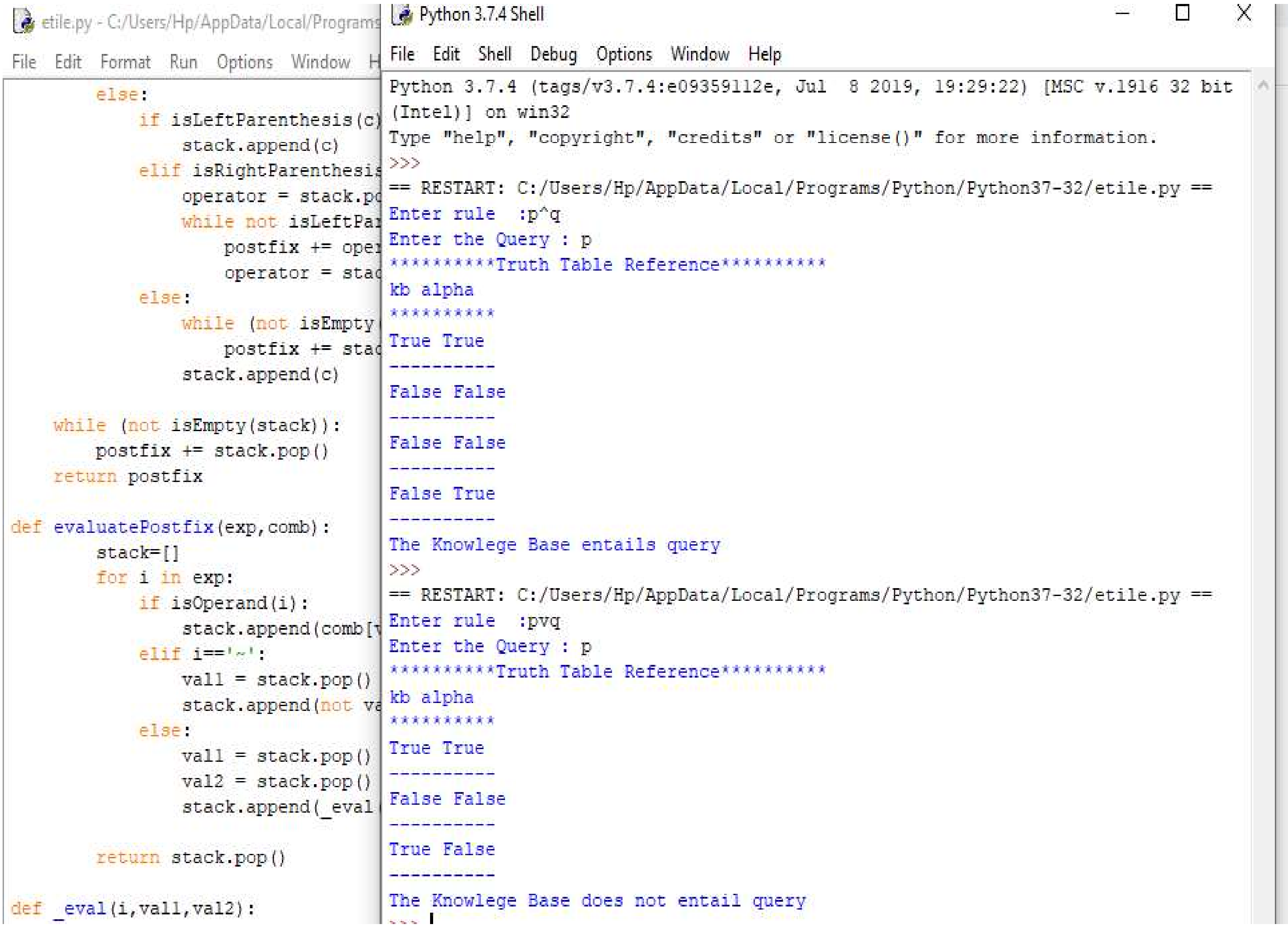
5. Implement Iterative deepening search to solve 8 puzzle problem.


8pp.py - C:/Users/Hp/AppData/Local/Programs/Python/Python
File Edit Format Run Options Window Help
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

```
        def IDDFS(self):
                def DLS(currentNode, depth):
                        if depth == 0:
                                return None
                        if currentNode.isSolve
                                return current
                        elif depth > 0:
                                for board in c
                                        nextNo
                                        if nex





                for depth in itertools.count()
                        visited = set()
                        startNode = Node(self.
                        #print(startNode.isSol
                        goalNode = DLS(startNo
                        if goalNode != None:
                                if goalNode.is
                                        return

startingBoard = [7,2,4,5,0,6,8,3,1]
myPuzzle = Puzzle(startingBoard)
mySolver = Solver(myPuzzle)
start = time.time()
goalSeq = mySolver.IDDFS()
end = time.time()
counter = -1 # starting state doesn't count a
for node in goalSeq:
        counter = counter + 1
        node.puzzle.printPuzzle()

print("Total number of moves: " + str(counter)
totalTime = end - start
print("Total searching time: %.2f seconds" %
```

```
6 3 5
0 7 8

1 4 2
0 3 5
6 7 8

1 4 2
3 0 5
6 7 8

1 0 2
3 4 5
6 7 8

0 1 2
3 4 5
6 7 8

1 4 2
6 0 3
7 8 5

1 4 2
6 3 0
7 8 5

1 4 2
6 3 5
7 8 0

1 4 2
6 3 5
7 0 8

1 4 2
6 3 5
0 7 8
```

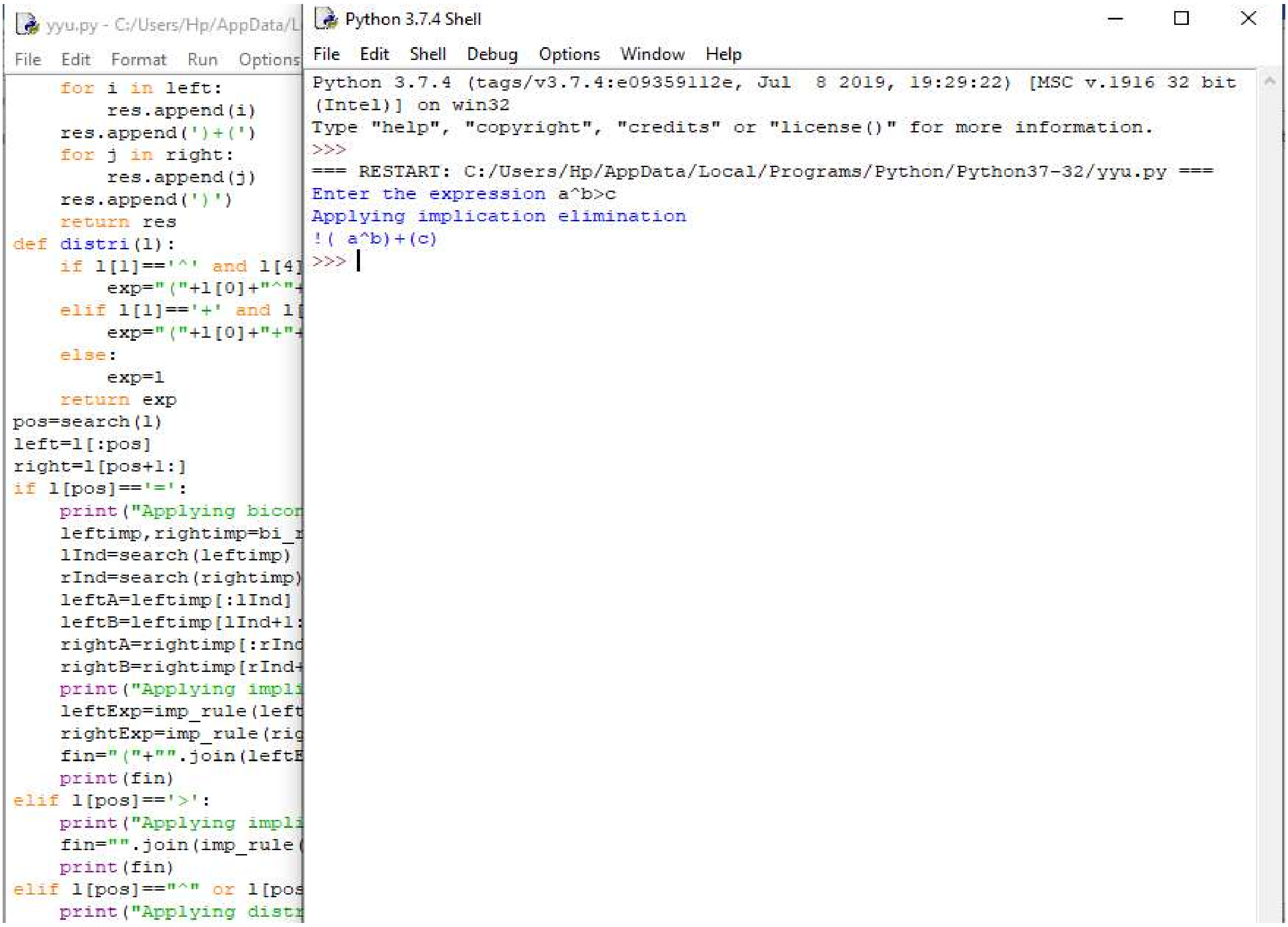
6. Create a knowledge based using prepositional logic and show that the given query entails the knowledge base or not


etile.py - C:/Users/Hp/AppData/Local/Programs
File Edit Format Run Options Window H
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

```
        else:
            if isLeftParenthesis(c)
                stack.append(c)
            elif isRightParenthesis
                operator = stack.po
                while not isLeftPar
                    postfix += oper
                    operator = stac
            else:
                while (not isEmpty(
                    postfix += stac
                stack.append(c)

    while (not isEmpty(stack)):
        postfix += stack.pop()
    return postfix

def evaluatePostfix(exp,comb):
        stack=[]
        for i in exp:
            if isOperand(i):
                stack.append(comb[v
            elif i=='~':
                val1 = stack.pop()
                stack.append(not va
            else:
                val1 = stack.pop()
                val2 = stack.pop()
                stack.append(_eval(

        return stack.pop()

def _eval(i,val1,val2):
```

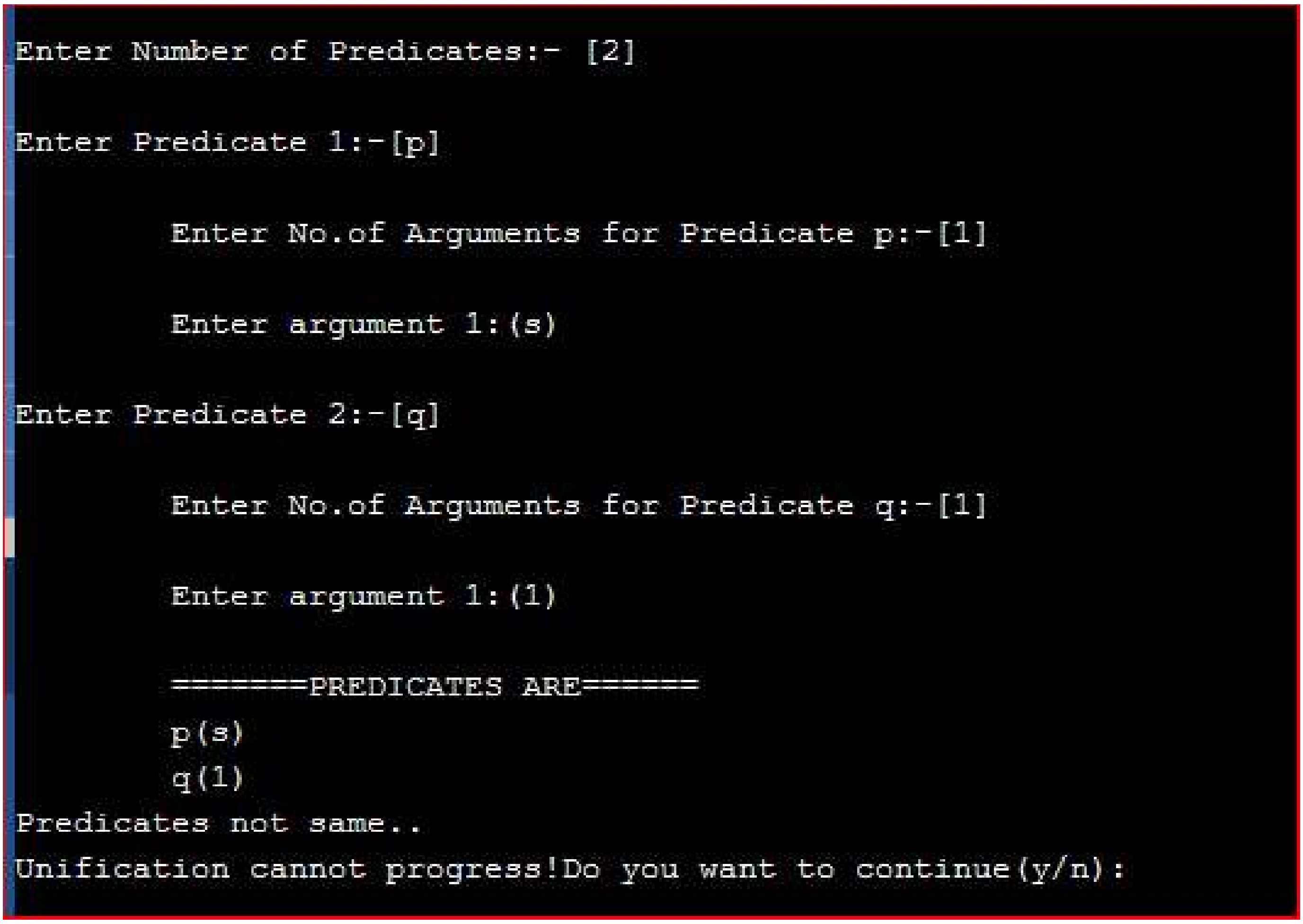
```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/etile.py ==
Enter rule  :p^q
Enter the Query : p
**********Truth Table Reference**********
kb alpha
**********
True True
----------
False False
----------
False False
----------
False True
----------
The Knowlege Base entails query
>>>
== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/etile.py ==
Enter rule  :pvq
Enter the Query : p
**********Truth Table Reference**********
kb alpha
**********
True True
----------
False False
----------
True False
----------
The Knowlege Base does not entail query
>>> 
```

7. **Implement Unification in First order logic.**


yyu.py - C:/Users/Hp/AppData/L
File Edit Format Run Options
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help

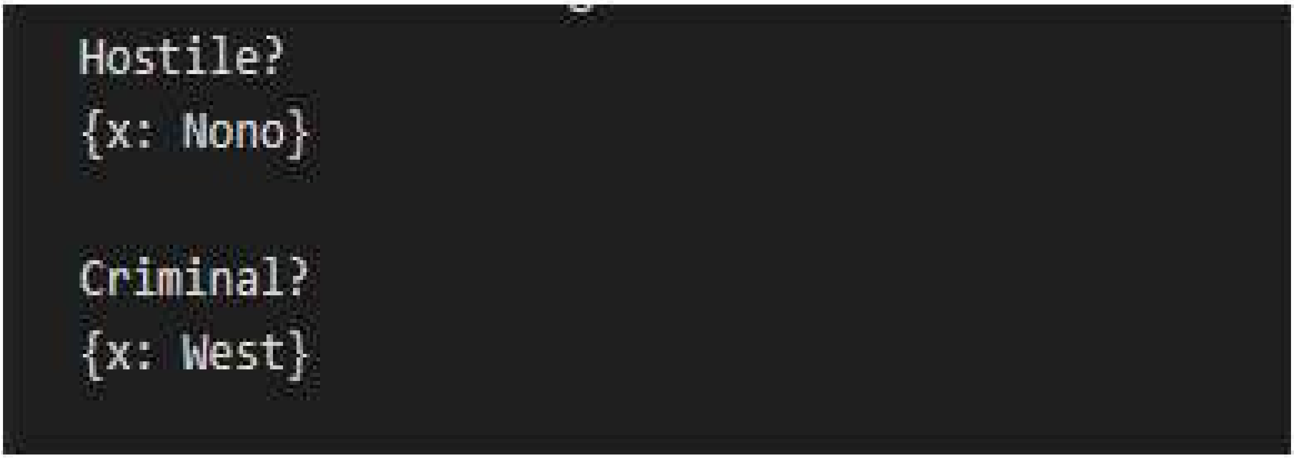
```
    for i in left:
        res.append(i)
    res.append(')+(')
    for j in right:
        res.append(j)
    res.append(')')
    return res
def distri(l):
    if l[1]=='^' and l[4]
        exp="("+l[0]+"^"+
    elif l[1]=='+' and l[
        exp="("+l[0]+"+"+
    else:
        exp=l
    return exp
pos=search(l)
left=l[:pos]
right=l[pos+1:]
if l[pos]=='=':
    print("Applying bicon
    leftimp,rightimp=bi_r
    lInd=search(leftimp)
    rInd=search(rightimp)
    leftA=leftimp[:lInd]
    leftB=leftimp[lInd+1:
    rightA=rightimp[:rInd
    rightB=rightimp[rInd+
    print("Applying impli
    leftExp=imp_rule(left
    rightExp=imp_rule(rig
    fin="("+"".join(leftE
    print(fin)
elif l[pos]=='>':
    print("Applying impli
    fin="".join(imp_rule(
    print(fin)
elif l[pos]=="^" or l[pos
    print("Applying distr
```

```
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul  8 2019, 19:29:22) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37-32/yyu.py ===
Enter the expression a^b>c
Applying implication elimination
!( a^b)+(c)
>>> 
```



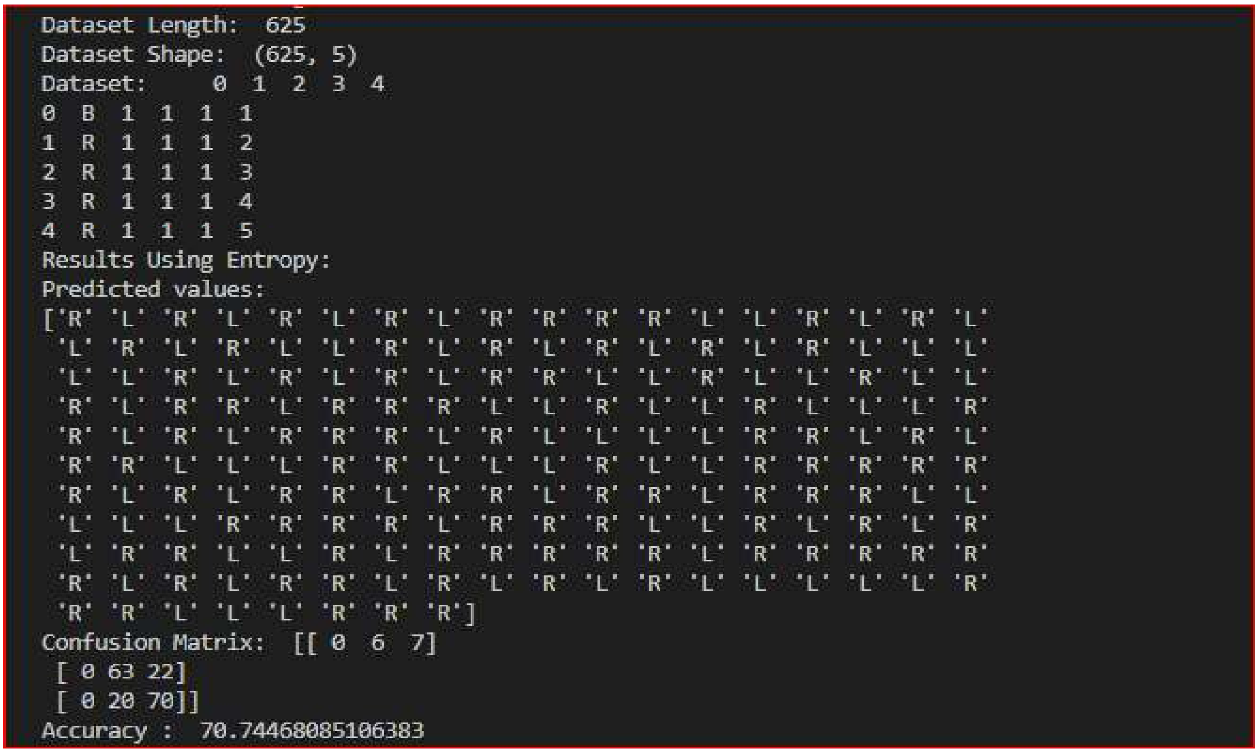
```
Enter Number of Predicates:- [2]

Enter Predicate 1:-[p]

        Enter No.of Arguments for Predicate p:-[1]

        Enter argument 1:(s)

Enter Predicate 2:-[q]

        Enter No.of Arguments for Predicate q:-[1]

        Enter argument 1:(1)

        =======PREDICATES ARE======
        p(s)
        q(1)
Predicates not same..
Unification cannot progress!Do you want to continue(y/n):
```

9. **Create a KB consisting of First order logic statements and prove the given query using forward reasoning.**

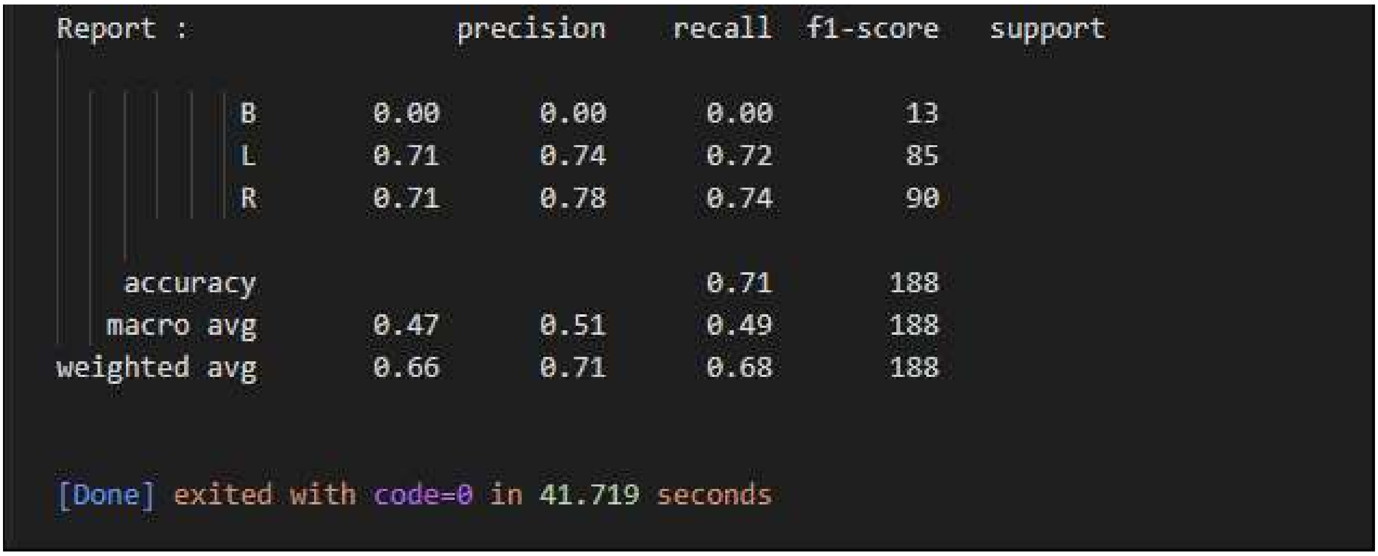


```
Hostile?
{x: Nono}

Criminal?
{x: West}
```

10. Demonstrate decision tree learning for a given set of training examples and test data.



```
Dataset Length:  625
Dataset Shape:  (625, 5)
Dataset:     0  1  2  3  4
0  B  1  1  1  1
1  R  1  1  1  2
2  R  1  1  1  3
3  R  1  1  1  4
4  R  1  1  1  5
Results Using Entropy:
Predicted values:
['R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L'
 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L'
 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:  [[ 0  6  7]
 [ 0 63 22]
 [ 0 20 70]]
Accuracy :  70.74468085106383
```



```
Report :                precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.71      0.74      0.72        85
           R       0.71      0.78      0.74        90

    accuracy                           0.71       188
   macro avg       0.47      0.51      0.49       188
weighted avg       0.66      0.71      0.68       188


[Done] exited with code=0 in 41.719 seconds
```