ARTIFICIAL INTELLIGENCE LAB PROGRAMS

OUTPUT

Tony J Mathew

1BM17CS119

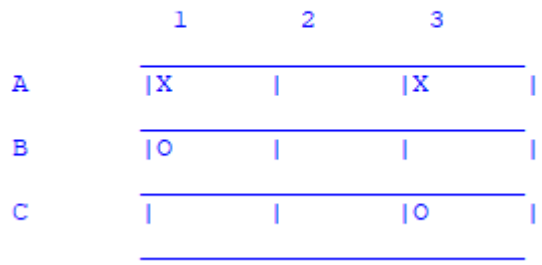
1. Implementation of Tic-Tac-Toe game

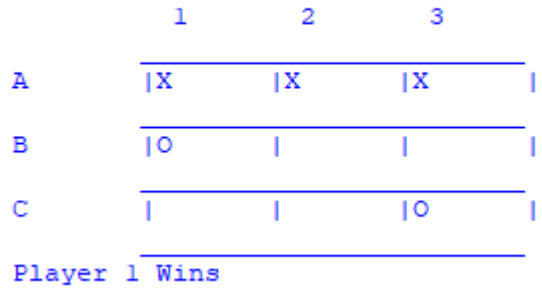
```
   1   2   3
 ___ ___ ___
A|   |   |   |
 ___ ___ ___
B|   |   |   |
 ___ ___ ___
C|   |   |   |
 ___ ___ ___
```

```
   1   2   3
 ___ ___ ___
A|X  |   |   |
 ___ ___ ___
B|   |   |   |
 ___ ___ ___
C|   |   |   |
 ___ ___ ___

Player 2:
Enter postion: C3
```

```
   1   2   3
 ___ ___ ___
A|X  |   |   |
 ___ ___ ___
B|   |   |   |
 ___ ___ ___
C|   |   |O  |
 ___ ___ ___
```

```
   1   2   3
 ___ ___ ___
A|X  |   |X  |
 ___ ___ ___
B|   |   |   |
 ___ ___ ___
C|   |   |O  |
 ___ ___ ___

Player 2:
Enter postion: B1
```






```
Player 1 Wins
>>>
```

2. Solving 8 puzzle problem

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

1 2 3
5 0 6
7 8 4

1 2 3
0 5 6
7 8 4

1 2 3
7 5 6
0 8 4

1 2 3
7 5 6
8 0 4

1 2 3
7 5 6
8 4 0

1 2 3
7 5 0
8 4 6

1 2 3
7 0 5
8 4 6

1 2 3
7 4 5
8 0 6

1 2 3
7 4 5
0 8 6

1 2 3
0 4 5
7 8 6

1 2 3
4 0 5
7 8 6

1 2 3
4 5 0
7 8 6

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

3. Implementation of vacuum cleaner agent

```
{'A': 0, 'B': 0}
Vacuum is randomly placed at Location A.
Moving to Location B...
{'A': 0, 'B': 0}
Performance Measurement: -1
>>>
================== RESTART: D:\BMSCE\2 Lab\AI\VacuumCleaner.py ===
{'A': 0, 'B': 1}
Vacuum cleaner randomly placed at Location B.
Location B is Dirty.
Location B has been Cleaned.
Moving to Location A...
{'A': 0, 'B': 0}
Performance Measurement: 0
>>>
================== RESTART: D:\BMSCE\2 Lab\AI\VacuumCleaner.py ===
{'A': 1, 'B': 1}
Vacuum is randomly placed at Location A.
Location A is Dirty.
Location A has been Cleaned.
Moving to Location B...
Location B is Dirty.
Location B has been Cleaned.
{'A': 0, 'B': 0}
Performance Measurement: 1
>>>
```

4. Implementation of A* search algorithm

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5. Implementing iterative deepening search to solve 8 puzzle problem

Screenshots not taken for middle part

Beginning

```
0 1 2
3 4 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
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
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 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
```

Ending

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

6 1 2
7 4 3
0 8 5

6 1 2
0 4 3
7 8 5

0 1 2
6 4 3
7 8 5

1 0 2
6 4 3
7 8 5

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

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

Total number of moves: 945
Total searching time: 4.61 seconds
```

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

```
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
>>>
==================== RESTART: D:\BMSCE\2 Lab\AI\entail.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
>>>
```

7. **Convert the given first order logic statement into conjunctive normal form (CNF)**

```
^ for and, + for or, ! for not, > for implies, = for biconditional
Enter the expressiona>(b^c)
Applying implication elimination
!(a)+((b^c))
>>>
==================== RESTART: D:\BMSCE\2 Lab\AI\FOLtoCNF.py ==========
^ for and, + for or, ! for not, > for implies, = for biconditional
Enter the expressiona>(b+c)
Applying implication elimination
!(a)+((b+c))
>>>
```

8. Implementation of unification in first order logic

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

Enter Predicate 1:-[p]

        Enter No.of Arguments for Predicate p:-[2]

        Enter argument 1:(a)

        Enter argument 2:(b)

Enter Predicate 2:-[p]

        Enter No.of Arguments for Predicate p:-[2]

        Enter argument 1:(c)

        Enter argument 2:(b)

        =======PREDICATES ARE======
        p(a,b)
        p(c,b)
        ======SUBSTITUTION IS======
        c/aDo you want to continue(y/n):
Process returned 10 (0xA)   execution time : 64.502 s
Press any key to continue.
```

9. Create a knowledge base consisting of first order logic statements and prove the query using forward reasoning

```
Hostile?
[{x: Nono}, {x: Jojo}, {x: Coco}]

Criminal?
[{x: West}]
```

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

```
========== RESTART: D:\BMSCE\2 Lab\AI\DecisionTree.py ==========
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
```