Ex : No : 5

Date :

MiniMax Algorithm.

program code :

```
from math import inf as infinity
from random import choice
import time
from os import system

HUMAN, COMP = -1, +1
board = [[0] * 3 for _ in range(3)]

def evaluate(state):
    if wins(state, COMP): return +1
    if wins(state, HUMAN): return -1
    return 0

def wins(state, player):
    win_state = [[state[0][0], state[0][1],
state[0][2], [state[1][0], state[1][1], state
[1][2], [state[2][0], state[2][1], state[2][2],
state[0][0], state[1][0], state[2][0],],
```

```
]
return [player, player, player] in win_state

def game_over(state):
    return [[x, y] for x, row in enumerate
(state) for y, cell in enumerate(row) if
cell == 0]

def set_move(x, y, player):
    if board[x][y] == 0:
        board[x][y] = player
        return True
    return False

def minmax(state, depth, player):
    if player == COMP: best = [-1, -1, -infinity]
    else: best = [-1, -1, +infinity]

    if depth == 0 or game_over(state):
        return [-1, -1, evaluate(state)]

    for x, y in empty_cells(state):
        state[x][y] = player
        score = minimax(state, depth-1,
-player)
```

```
if player == COMP:
    if score[2] > best[2]: best = score.
else:
    if score[2] < best[2]: best = score.
return best.

def clean():
    system('cls' if 'win' in platform.system().
lower() else 'clear')

def render():
    chars = {HUMAN: 'x', COMP: 'o', 0: ''}
    print('\n' + '---'* 3)
    for row in board:
        print('|' + '|'.join([chars[cell] for
cell in row]) + '|')
        print('---' * 3)

def ai_turn():
    move = minmax(board, len(empty-
cells(board)), COMP)
    set_move(move[0], move[1], COMP)
    time.sleep(1)
```

```
def human_turn():
    move = -1
    while move not in range(1,10):
        try:
            move = int(input('Enter move (1-9): '))
            x, y = divmod(move-1, 3)
            if not set_move(x, y, HUMAN):
                print('Invalid move! Try again.')
        except (ValueError, IndexError):
            print('Invalid Input!')

def main():
    clean()
    render()
    while len(empty_cells(board)) > 0 and not
game_over(board):
        human_turn()
        if game_over(board): break
        ai_turn()
        render()
    if wins(board, HUMAN): print('You win!')
    elif wins(board, COMP): print('You lose!')
```

```
else: print ('Draw!')

if __name__ == '__main__':
    main().
```

output:

```
choose X or O
chosen: X
First to start? [y/n]: y
Human turn [X].
 . . . . . .
|  ||   ||   |
|  ||   ||   |
|  ||   ||   |
 - - - - - - -
use numpad (1..9): 4.
computer turn [O].
```

Result:

Thus the program for Mini Max Algorithm is successfully executed & the output is verified.