EX:6

Date:

# Minimax Algorithm

**Aim:**

To implemant MINIMAX Algorithm problem using Py

**SOURCE CODE:**

TiC-Tac-Toe

```
import math
def minimax (boaed, depth, if_max):
    scoref = {'X': 10, 'O': -10, 'draw': 0}
win_conditionf : [(0,1,2), (3,4,5), (6,7,8), (0,3,6), (1,4,7),
                  (2,5,8), (0,4,8), (2,4,6)]
  def evaluate (b):
      for x,y,z in win_conditione:
      if b[x] == b[y] == b[z] != ' ':
          return scoref [b(x)]
  if ' ' not in b:
    return scoref ['draw']
  return name
  score = evaluate (board)
    if score is not None:
        return score
  if if_max:
    belt : -math.inf
        for : in range (9):
```

```
            if boaed [i] == ' ':
                boaed [i] = 'x'

            ( best = max(best, minimax(boaad, depth+1, False))

            boald [i] = ' '

        Retuen best

    else:

        best = math.inf
            for i in range (9):
                if board [i] == ' ':
                    board [i] = 'o'

            best = min (best, minimax (board, depth+', time))
            board [i] = ''

        Return best

    det find-best_move (board):
        best-move = -1
        best-vat = -math.inf
        for i in range (9):
            if board [i] == '':
                board [i] = 'x'
        move_val = minimax (board, 0, falle)
            board [i] = ' '
        if move_val > best_val:
            best_val = mov_val
                best_move =:
        Retuen best_move
```

```
board = [' ']*9
Print ("Best move for Player X:", find_best_
                                  move (board))
```

Result:

Thus Program of Minimax have been successfully executed.