# Lab - 1

* program for Tic - Tac - Toe .

```
board = { 1: '  ', 2: '  ', 3: '  ',
          4: '  ', 5: '  ', 6: '  ',
          7: '  ', 8: '  ', 9: '  '}

def print_board (board) :
    print (board [1] + '|' + board [2] + '|' + board [3])
    print ('-+-+-')
    print (board [4] + '|' + board[5] + '|' + board [6])
    print ('-+-+-')
    print (board [7] + '|' + board [8] + '|' + board [9])
    print ('\n')

def space_free (pos):
    return board [pos] == ' '

def check win ():
    win-conditions = [
        (1,2,3), (4,5,6), (7,8,9),
        (1,4,7) (2,5,8), (3,6,9),
        (1,5,9) , (3,5,7)
    ]
    for a,b,c in win_conditions:
        if board[a] == board[b] == board[c] != ' ':
            return True
    return False

def check_draw ():
    return all(board [key] != ' ' for key in board).
```

```
def insert_letter(letter, position):
    if space_free(position):
        board[position] = letter
        print_board(board).

        if check_win():
            if letter != 'X':
                print('Bot wins!')

            else:
                print('You win!'):

            return True

        elif check_draw():
            print('Draw!')
            return True
        return False

    else:
        print('Position taken, please pick of a different position')
        position = int(input('Enter new position:'))
        return insert_letter(letter, position)

player = 'O'
bot = 'X'

def player_move():
    position = int(input('Enter position for O:")).
    return insert_letter(player, position)

def comp_move():
    best_score = -1000
    best_move = 0
    for key in board.keys():
        if space_free(key):
            board[key] = bot
            score = minimax(board, False)
            board[key] = ' '
            if score > best_score:
                best_score = score
                best_move = key

    insert_letter(bot, best_move)

def minimax(board, is_maximizing):
    if check_win():
        return 1 if board[1] == bot else -1
    elif check_draw():
        return 0

    if is_maximizing:
        best_score = -1000
        for key in board.keys():
            if space_free(key):
                board[key] = bot
                score = minimax(board, False)
                board[key] = ' '
                best_score = max(best_score, score)
        return best_score
    else:
        best_score = 1000
        for key in board.keys():
            if space_free(key):
                board[key] = player
                score = minimax(board, True)
            board[key] = ' '
            best_score = min(best_score, score)

        return best_score


while True:
    print_board(board)
    if check_draw() or check_win():
        break
    comp_move()
    if check_draw() or check_win():
        break
    player_move().
```

Output:

```
Enter position for O: 5

X |   |
-+-+-
  | O |
-+-+-
  |   |

Enter position for O: 3

X | X | O
-+-+-
  | O |
-+-+-
  |   |

Enter position for O: 7

X | X | O
-+-+-
X | O |
-+-+-
O |   |
```

[illegible] 04.10