Sub : AI - Lab test 1.

Name : Prathibha.R.
USN : 1BM18CS074
Class : 5th "B"

Pseudocode

main functions that we would require :-

① creating a board.
② checking for empty spaces on the board.
③ Select a place for the player.
④ Check if the marks are horizontally done.
⑤ Check if the marks are vertically aligned.
⑥ check if the marks are diagonally in a row.
⑦ Evaluation of the winner or if there is a tie.
⑧ Main function to start the game.

Since we have computer vs computer, we donot require user's input.

```
{importing all the required libraries}

def create_board():
  return (np.array([[0,0,0],
                    [0,0,0],
                    [0,0,0]]))

def check_for_space(board):
    l = []
    for i in range(len(board)):
        for j in range(len(board)):
            if board[i][j] == 0:
                l.append((i,j))
    return (l)
```

```
def random_place (board, player)
      selection = check_for_space (board)
      current = random, choice (selection)
      board [current] = player.
       return board.

  def row_checking (board, player):
     for x in range (len[board)];
          win = True
          for j in range(len[board]):
               if board [x, j] != player:
                     win = False.
                     continue
          if win == True:
               return (win)
       return(win)
    def col_checking (board, player):
      for i in range[len[board)]:
           win = True
          for j in range[len [board]]:
             if board [j][i] = player:
               win = False
                continue.
         if win == True;
              return(win)
        return (win)
     def diag_checking (board, player):
            win = True.
              y = 0.
      for x in range (len[board]):
           if board [x, x] != player:
                 win = False.
         if win:
             return win.
        win = True.
```

Prathul

```
if win:
    for i in range(len(board)):
        j = len[board] - 1 - i
        if board[i,j] ! = player:
            win = False
    return win.
def what_is_the_final_result(board):
    winner = 0.
    for player in [1,2]:
        if (row-checking (board, player) or
            col-checking (board, player) or dia-checking
            (board, player)):
            winner = player
    if np.all(board != 0) and winner == 0:
        winner = -1
    return winner
def play-now():
    board, winner, counter = create-board(),0,1
    Print(board).
    Sleep(2)
    while winner == 0:
        for player in [1,2]:
            board = random-place(board, player)
            Print("Board after" + str(counter) + 'move')
            print(board).
            sleep(2)
            Counter += 1.
            winner = what-is-the-final-result(board)
            if winner != 0;
                break;
        return (winner)
Print("winner is :" + str(play-now()))
```

Prathi