```
init()
    initialise variables ( board, weights, ai, player)

move(character)
  "if the cell is empty, assign the character to
that cell on the board and return true else return false
  if board[row][col] == ' '
      board[row][col] = character
      weights[row][col] = 0
      return true
  return false

display( move_type)
    if move_type == 'cpu'
          print ("CPUs move")
    for i in range(3)
        for j in range (3)
            print (board[i][j])

compare_line(sl, ch)
    return ' ' in sl and sl.count(ch) == 2

get_position()
   max_val = max([max(x) for x in weights])
   positions = [(i, weights[i].index(max_value)) for i in range(3)
                    if max_val in weights[i]]
  return positions.

has_tied()
     for row in board
           if ' ' in rows return false
     return true
```

```
attacking_position(ch)
    default = ' '
    for i in range(3)
     col = [board[0](i], board[1][i], board[2][i]]
       if compare-line(col, ch) return (col.index(default), i)
    diagonal1 = [board[0][0], board[1][1], board[2][2]]
    diagonal2 = [board[0][2], board[1][1], board[2][0]]
    if compare_line(diag1, ch) ~~return~~
        return (diag1.index(default), diag1.index(default))
    elif compare_line(digonal2, ch)
        return (dig2.index(default), 2 - diag2.index(default))
    return False

ai_move()
    pos, f = attacking_position(ch = ai), False
    if pos! = False
        (row, col), f = pos, True
    else
      pos = attacking_position(ch=player)
      if pos! = False  (row, col) = pos
      else   (row, col) = r.choice(get_position())
    move(row, col, ai)
    return f
```

```
run()
    global ai, player
    end, tied, move_type = False, False, None
    ~~print~~ display()
    ch = input('X or O')
    if ch=='O' ai,player = player, ai
    while (True)
        if tied
            print("Tied") return
        elif end
            print(move_type+'has won')
        move_type ='player'
        r=int(input("Row"))
        c = int(input("Column"))
        if not move(r-1, c-1, player)
            print('Enter correct position')
        else
            display(move_type)
            tied =has_tied()
            if tied: continue
            move_type = 'cpu'
            end= ai_move()
            display (move_type)
            tied = has_tied()

main()
    run()
    f='Y'
    while(f=='Y' or f=='y')
        f=input('Play again?')
        init()
        if f=="Y" or f=='y' run()
```