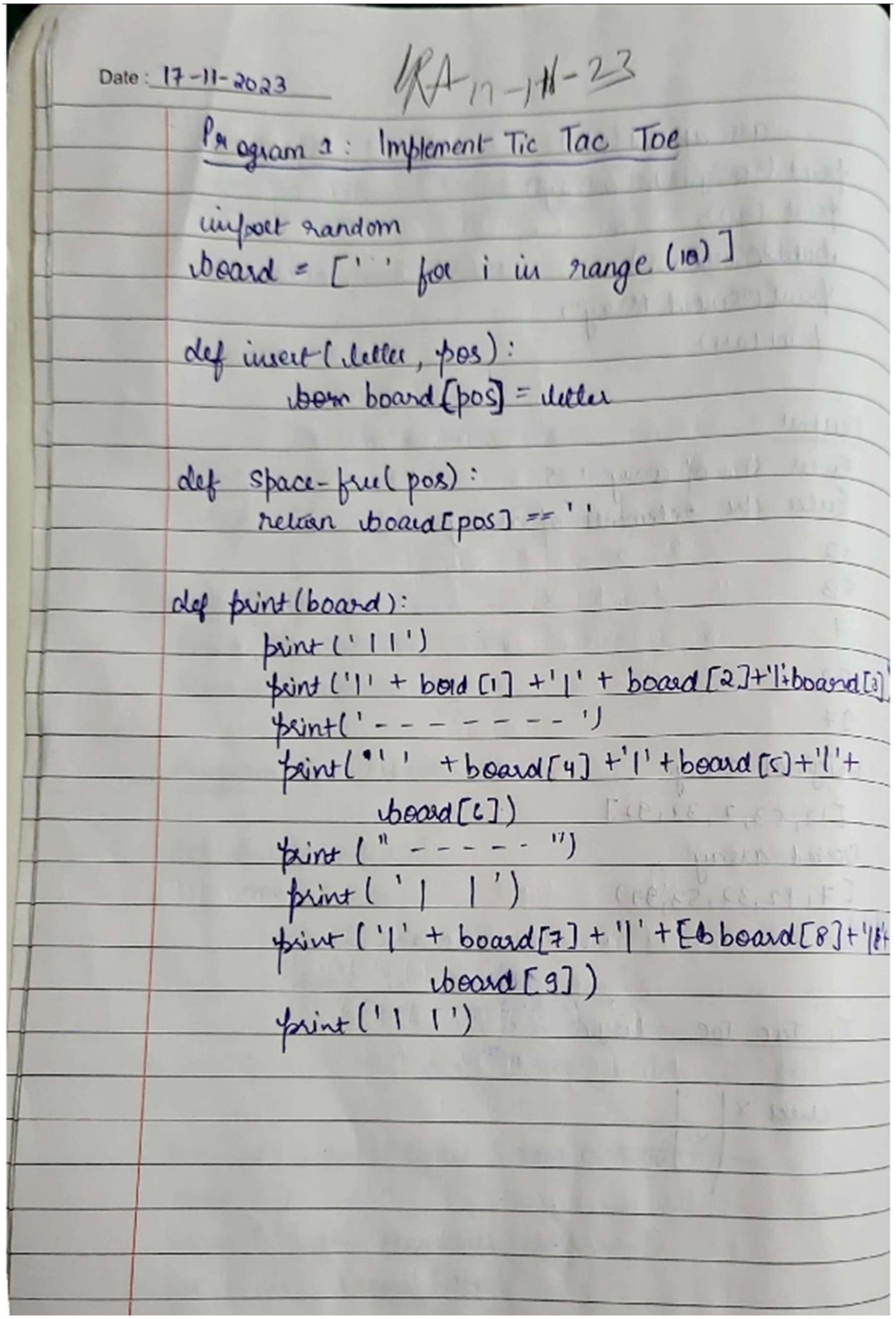
Table of Contents

Program 1: Implement Tic Tac Toe

Code:

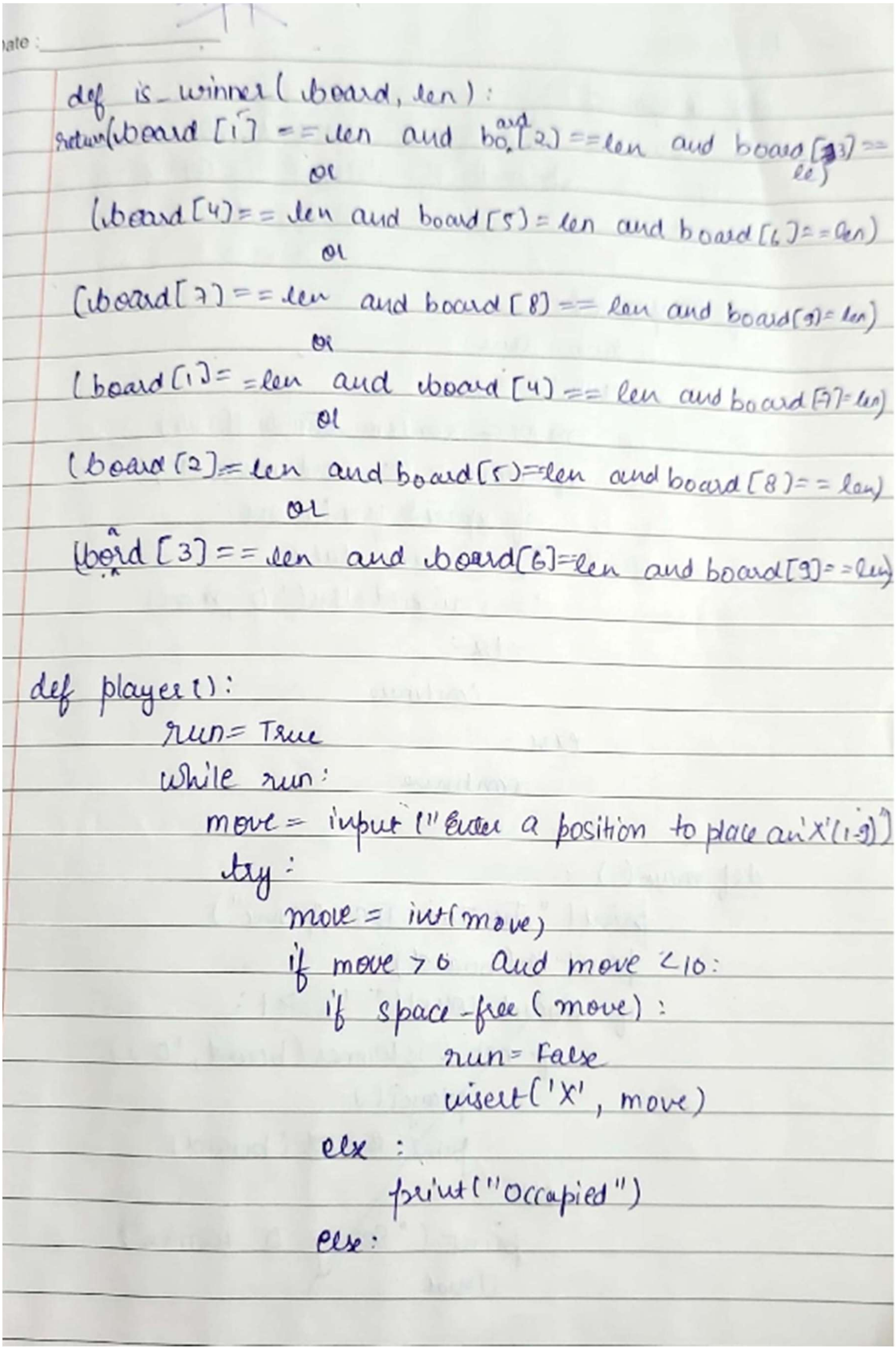
```
board = [' ' for x in range(10)]

def insertLetter(letter, pos):
    board[pos] = letter

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

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

def isWinner(bo, le):
    return (bo[7] == le and bo[8] == le and bo[9] == le) or (bo[4] == le 
and
    bo[5] == le and bo[6] == le) or (bo[1] == le and bo[2] == le and 
bo[3] == le) or (bo[1] == le and
    bo[4] == le and bo[7] == le) or (
    bo[2] == le and bo[5] == le and bo[8] == le) or (
    bo[3] == le and bo[6] == le and bo[9] == le) or (
    bo[1] == le and bo[5] == le and bo[9] == le) or (bo[3] ==
    le and bo[5] == le and bo[7] == le)

def playerMove():
    run = True
    while run:
        move = input('Please select a position to place an \'X\' (1-9): 
')
        try:
            move = int(move)
            if move > 0 and move < 10:
                if spaceIsFree(move):
                    run = False
                    insertLetter('X', move)
                else:
                    print('Sorry, this space is occupied!')
            else:
                print('Please type a number within the range!')
        except:
            print('Please type a number!')

def compMove():
    possibleMoves = [x for x, letter in enumerate(board) if letter == ' '
and x
    != 0]
    move = 0
    for let in ['O', 'X']:
        for i in possibleMoves:
            boardCopy = board[:]
            boardCopy[i] = let
            if isWinner(boardCopy, let):
                move = i
                return move
    cornersOpen = []
    for i in possibleMoves:
        if i in [1, 3, 7, 9]:
            cornersOpen.append(i)
    if len(cornersOpen) > 0:
        move = selectRandom(cornersOpen)
        return move
    if 5 in possibleMoves:
        move = 5
        return move
    edgesOpen = []
    for i in possibleMoves:
        if i in [2, 4, 6, 8]:
            edgesOpen.append(i)
    if len(edgesOpen) > 0:
        move = selectRandom(edgesOpen)
        return move

def selectRandom(li):
    import random
    ln = len(li)
    r = random.randrange(0, ln)
    return li[r]

def isBoardFull(board):
    if board.count(' ') > 1:
        return False
    else:
        return True

def main():
    print('Welcome to Tic Tac Toe!')
    printBoard(board)
    while not (isBoardFull(board)):
        if not (isWinner(board, 'O')):
            playerMove()
            printBoard(board)
        else:
            print('Sorry, O\'s won this time!')
            break
    if not (isWinner(board, 'X')):
        move = compMove()
        if move == 0:
            print('Tie Game!')
        else:
            insertLetter('O', move)
            print('Computer placed an \'O\' in position', move, ':')
            printBoard(board)
    else:
        print('X\'s won this time! Good Job!')
    if isBoardFull(board):
        print('Tie Game!')

while True:
    answer = input('Do you want to play again? (Y/N)')
    if answer.lower() == 'y' or answer.lower() == 'yes':
        board = [' ' for x in range(10)]
        print('-----------------------------------')
        main()
    else:
        break
```



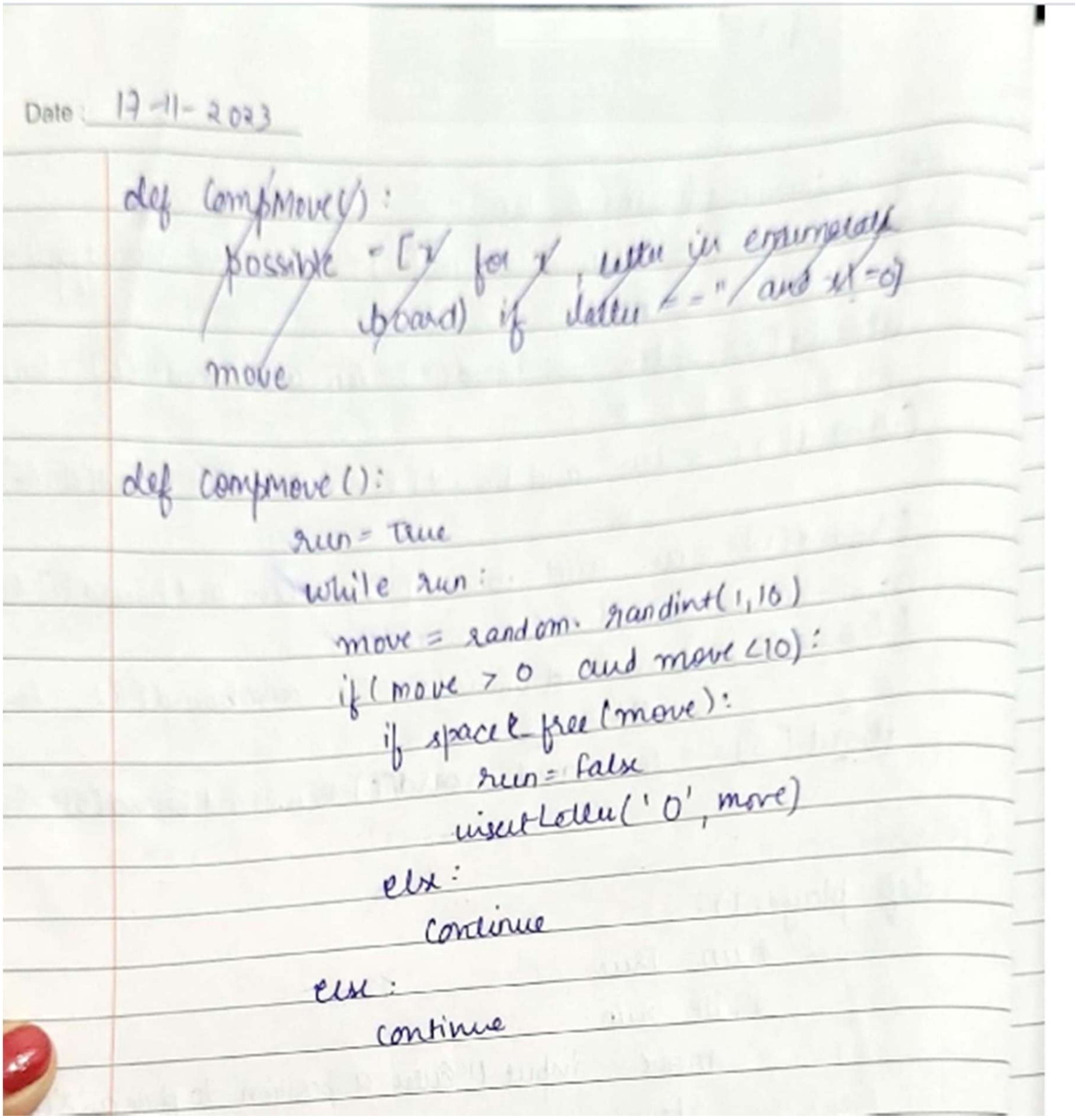
Date : 17-11-2023 RA 17-11-23

Program 2 : Implement Tic Tac Toe

```
import random
board = ['' for i in range (10)]

def insert(letter, pos):
        ~~bom~~ board[pos] = letter

def space-free(pos):
        return board[pos] == ''

def print(board):
        print('||')
        print('|' + bord [1] +'|' + board [2]+'|'+board[3]
        print('- - - - - - - - ')
        print('|' +board[4] +'|'+board[5]+'|'+
                board[6])
        print (" - - - - - ")
        print ( '|  |')
        print ('|' + board[7] + '|' +[b board[8]+'|'+
                board[9] )
        print('|  |')
```

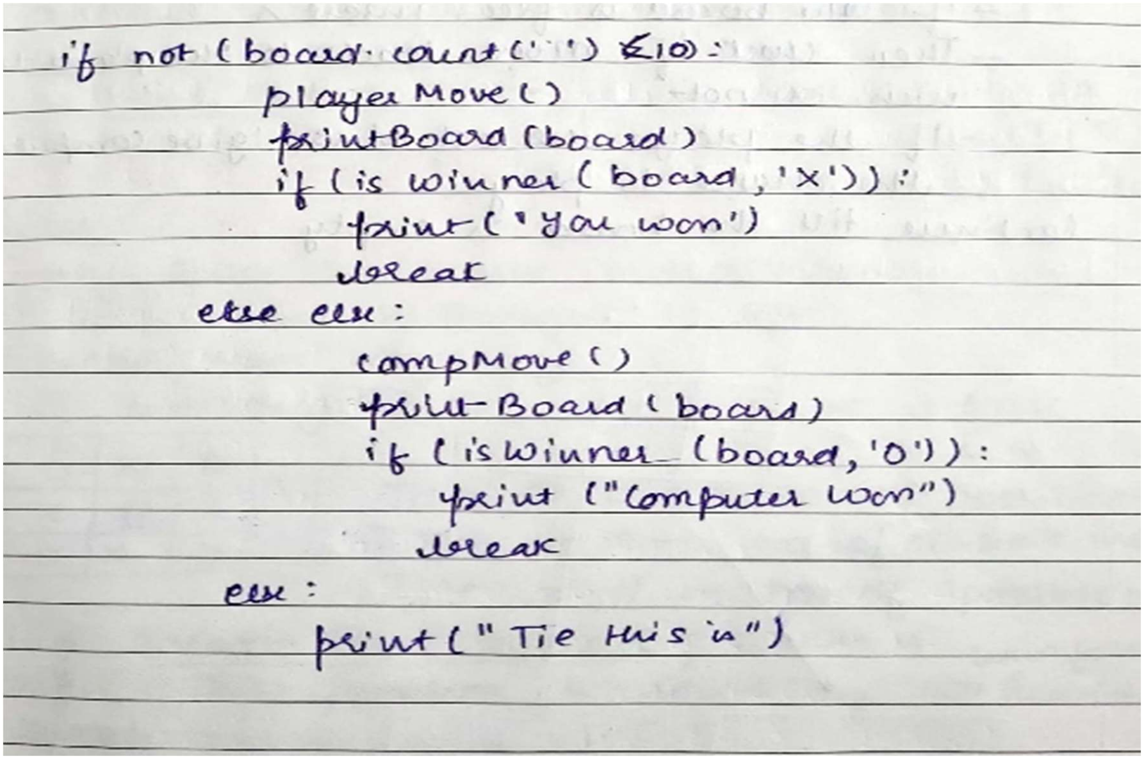


```
def is_winner( board, len ):
return(board [1] == len and board[2]==len and board[3] == le)
            or
  (board[4]== len and board[5] = len and board[6]==len)
            or
 (board[7]== len  and board[8] == len and board[9]= len)
            or
 (board[1]= =len and board [4] == len and board[7]= len)
            or
 (board[2]= len and board[5]=len and board[8]= = len)
            or
 (board [3] == len and board[6]=len and board[9]==len)


def player():
        run= True
        while run:
            move= input("Enter a position to place an'X'(1-9)")
            try:
                move = int(move)
                if move >0  and move <10:
                    if space-free(move):
                            run= False
                            insert('X', move)
                    else :
                            print("Occapied")
                else:
```

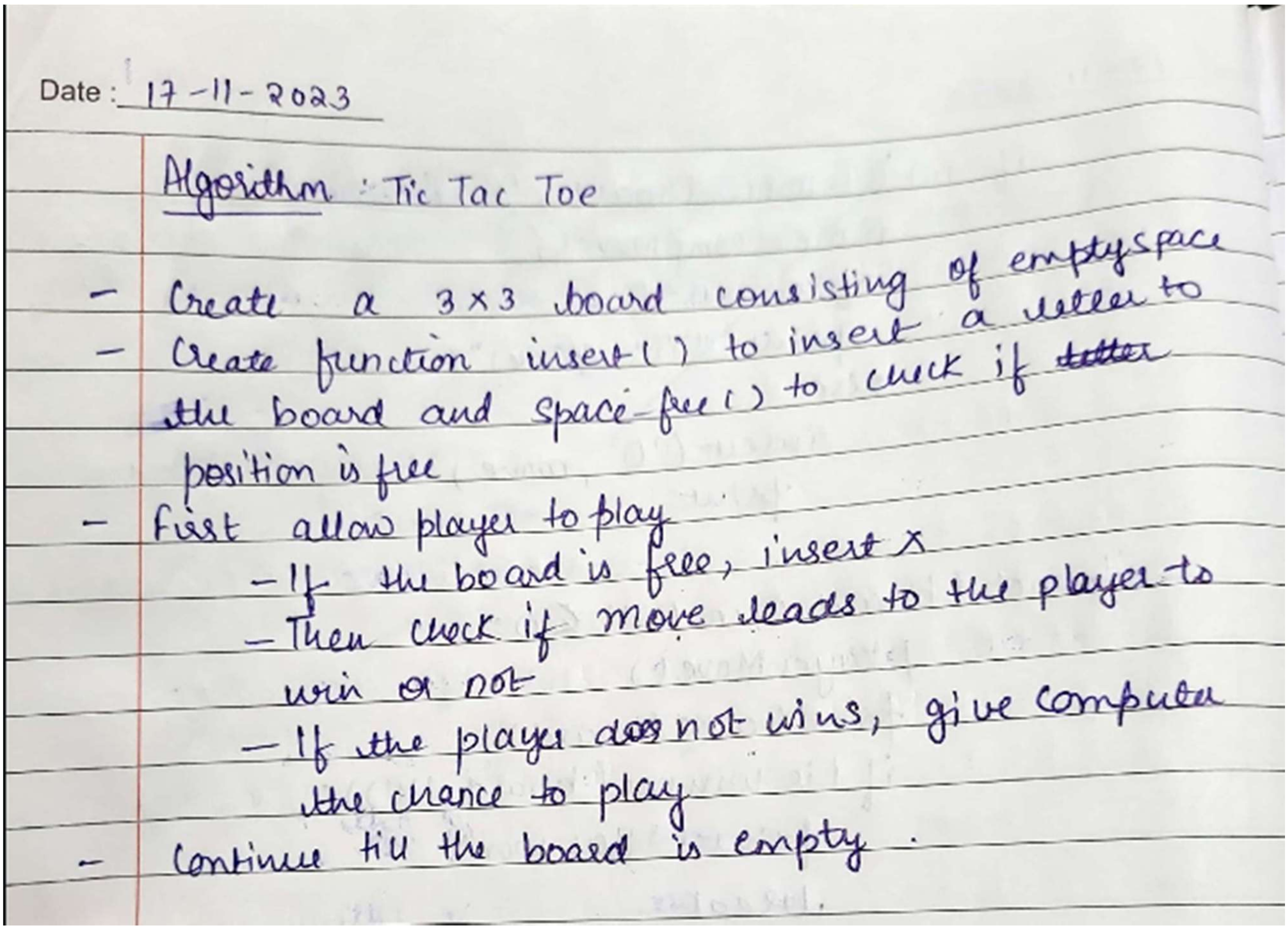


Date : 17-11-2023

```
~~def compMove():
    possible = [x for x, letter in enumerate
              (board) if letter == '' and x != 0]
    move~~

def compmove():
        run = True
        while run:
          move = random. randint(1,10)
          if(move > 0 and move <10):
            if space_free(move):
                run = false
                insertLetter('0', move)
            else:
                continue
        else:
            continue
```



```
if not (board. count(' ') <10):
        playerMove()
        printBoard(board)
        if(is winner(board, 'X')):
            print('You won')
            break
    ~~else~~ else:
            compMove()
            print-Board(board)
            if(isWinner_(board, '0')):
                print("Computer won")
                break
    else:
        print("Tie this is")
```

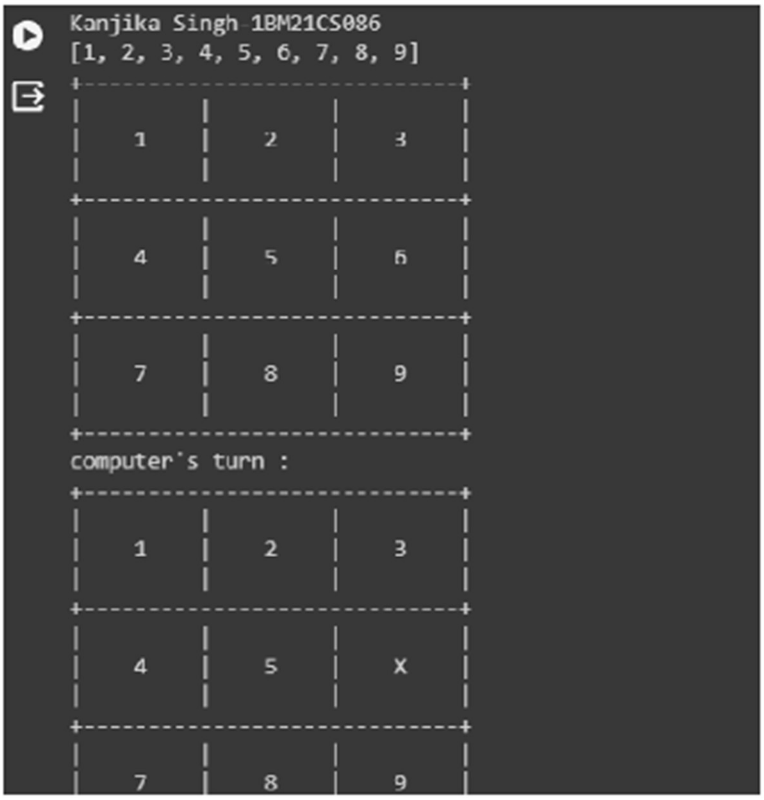


Date : 17-11-2023

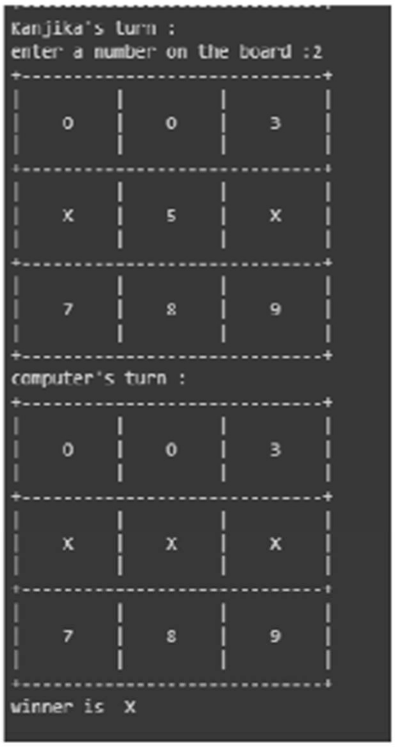
Algorithm : Tic Tac Toe

- Create a 3 × 3 board consisting of emptyspace
- Create function insert( ) to insert a letter to the board and space-free( ) to check if ~~letter~~ position is free
- First allow player to play
    - If the board is free, insert x
    - Then check if move leads to the player to win or not
    - If the player does not wins, give computer the chance to play
- Continue till the board is empty .

Output:



```
Kanjika Singh 1BM21CS086
[1, 2, 3, 4, 5, 6, 7, 8, 9]
+-----------------------------+
|         |         |         |
|    1    |    2    |    3    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    4    |    5    |    6    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    9    |
|         |         |         |
+-----------------------------+
computer's turn :
+-----------------------------+
|         |         |         |
|    1    |    2    |    3    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    4    |    5    |    X    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    9    |
```



```
Kanjika's turn :
enter a number on the board :2
+-----------------------------+
|         |         |         |
|    O    |    O    |    3    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    X    |    5    |    X    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    9    |
|         |         |         |
+-----------------------------+
computer's turn :
+-----------------------------+
|         |         |         |
|    O    |    O    |    3    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    X    |    X    |    X    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    9    |
|         |         |         |
+-----------------------------+
winner is  X
```