```
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()
```

```
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] and board[a] != ' ':
            return True
    return False
```

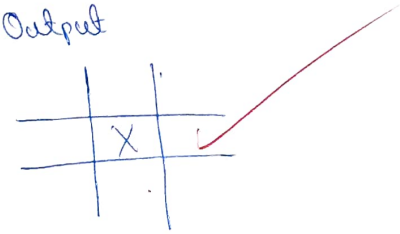
```
def check_draw():
    return all(space != ' ' for space in board.values())
```

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

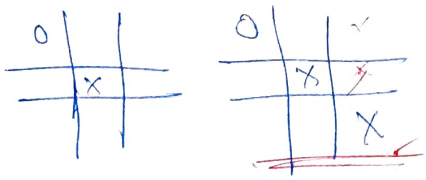
```
    elif check_draw():
        print('draw!')

    elif check_win():
        print(f'{letter} wins!')

    return

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

player = 'O'
bot = 'X'

def player_move():
    position = int(input('Enter position for O (1-9): '))
    insert_letter(player, position)

def comp_move():
    best_score = -1000
    best_move = 0
    for key in board.keys():
        if board[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 is_maximizing else -1
    if check_draw():
        return 0

    if is_maximizing:
        best_score = -1000
        for key in board.keys():
            if board[key] == ' ':
```

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

while not check_win() and not check_draw():
    comp_move()
    if not check_win() and not check_draw():
        player_move()
```

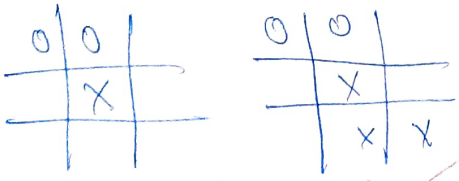
Output



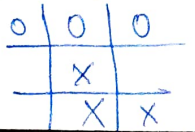
Enter position for o (1-9): 1



Enter position for o (1-9): 2



Enter position for x (1-9): 3



O wins!

04.10.24