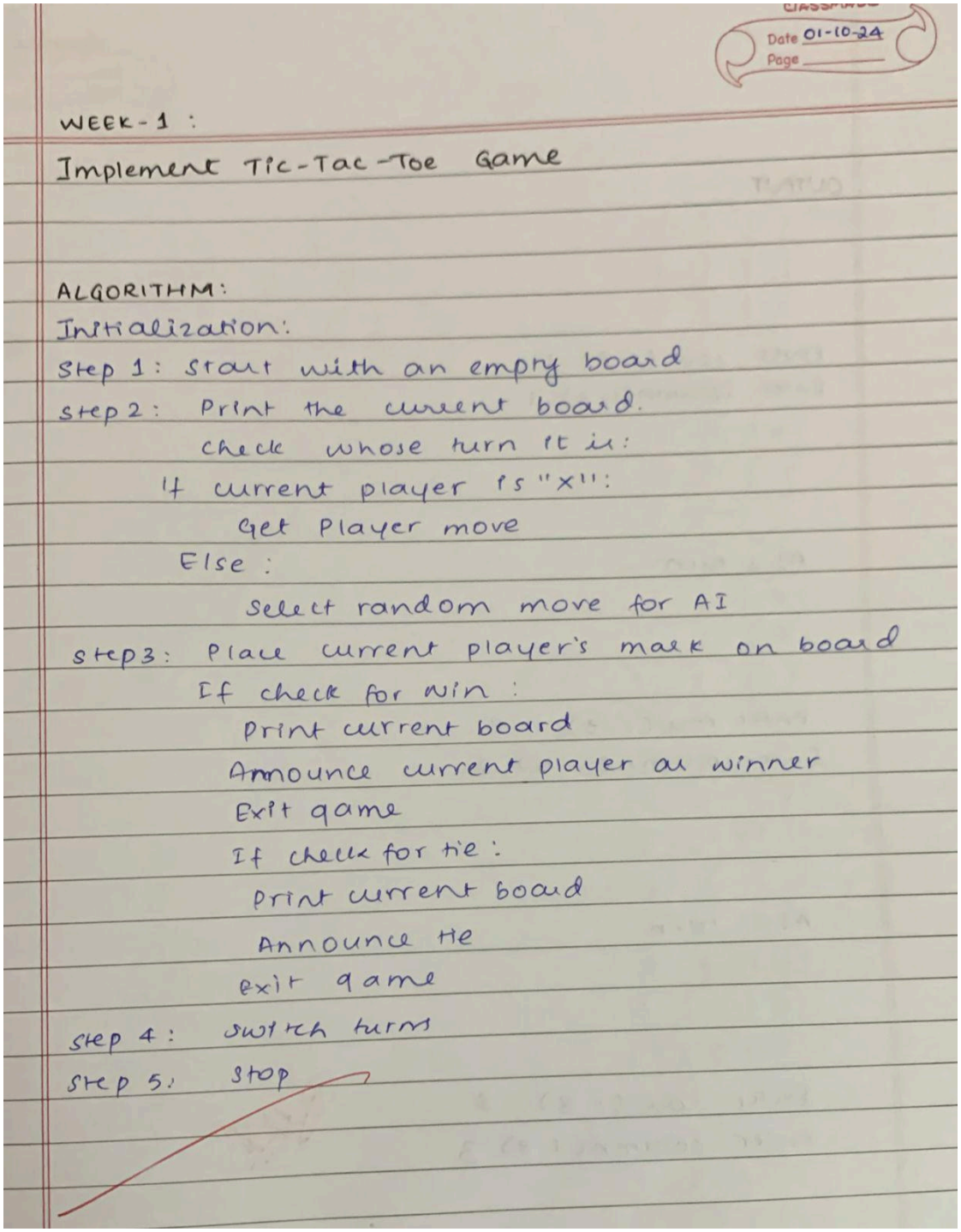
01/10/2024 LAB 01- TIC TAC TOE GAME

ALGORITHM/PSEUDOCODE-



Date 01-10-24
Page

WEEK-1 :

Implement Tic-Tac-Toe Game

ALGORITHM:

Initialization:

Step 1: Start with an empty board

step 2: Print the current board.

Check whose turn it is:

If current player is "X":

Get Player move

Else :

Select random move for AI

step3: Place current player's mark on board

If check for win :

Print current board

Announce current player as winner

Exit game

If check for tie :

Print current board

Announce tie

Exit game

step 4: switch turns

step 5: stop

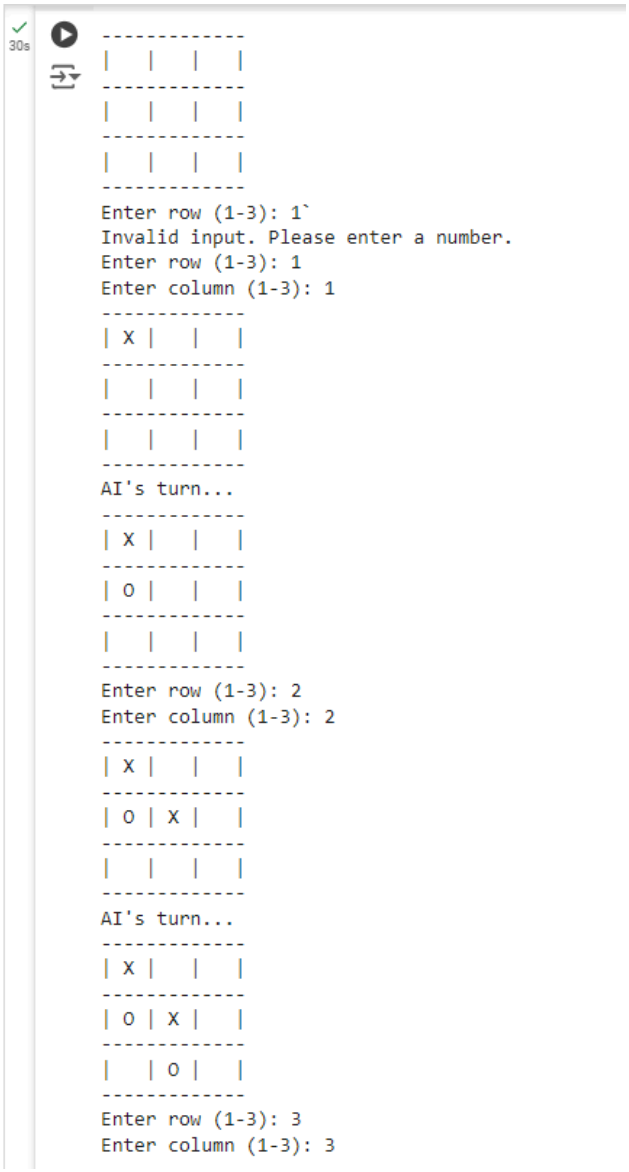
CODE/INPUT-

```
import random

def print_board(board):
  """Prints the Tic Tac Toe board."""
  print("-------------")
  for row in board:
    print("|", end="")
    for cell in row:
      print(" " + cell + " |", end="")
    print("\n-------------")


def check_win(board, player):
  """Checks if the player has won."""
  # Check rows
  for row in board:
    if all(cell == player for cell in row):
      return True

  # Check columns
  for col in range(3):
    if all(board[row][col] == player for row in range(3)):
      return True

  # Check diagonals
  if all(board[i][i] == player for i in range(3)):
    return True
  if all(board[i][2 - i] == player for i in range(3)):
    return True

  return False

def is_board_full(board):
  """Checks if the board is full."""
  for row in board:
```

```
    for cell in row:
      if cell == " ":
        return False
  return True


def get_player_move(board):
  """Gets the player's move."""
  while True:
    try:
      row = int(input("Enter row (1-3): ")) - 1
      col = int(input("Enter column (1-3): ")) - 1
      if 0 <= row <= 2 and 0 <= col <= 2 and board[row][col] == " ":
        return row, col
      else:
        print("Invalid move. Try again.")
    except ValueError:
      print("Invalid input. Please enter a number.")


def get_ai_move(board):
  """Gets the AI's move using a simple random strategy."""
  empty_cells = []
  for row in range(3):
    for col in range(3):
      if board[row][col] == " ":
        empty_cells.append((row, col))
  if empty_cells:
    return random.choice(empty_cells)
  return None


def main():
  """Main function to run the game."""
  board = [[" " for _ in range(3)] for _ in range(3)]
  current_player = "X"
  ai_player = "O"
```

```
  while True:
    print_board(board)

    if current_player == "X":
      row, col = get_player_move(board)
    else:
      print("AI's turn...")
      row, col = get_ai_move(board)

    board[row][col] = current_player

    if check_win(board, current_player):
      print_board(board)
      print(current_player + " wins!")
      break
    elif is_board_full(board):
      print_board(board)
      print("It's a tie!")
      break

    current_player = "O" if current_player == "X" else "X"


if __name__ == "__main__":
  main()
3
```

OUTPUT-



```
-------------
|   |   |   |
-------------
|   |   |   |
-------------
|   |   |   |
-------------
Enter row (1-3): 1`
Invalid input. Please enter a number.
Enter row (1-3): 1
Enter column (1-3): 1
-------------
| X |   |   |
-------------
|   |   |   |
-------------
|   |   |   |
-------------
AI's turn...
-------------
| X |   |   |
-------------
| O |   |   |
-------------
|   |   |   |
-------------
Enter row (1-3): 2
Enter column (1-3): 2
-------------
| X |   |   |
-------------
| O | X |   |
-------------
|   |   |   |
-------------
AI's turn...
-------------
| X |   |   |
-------------
| O | X |   |
-------------
|   | O |   |
-------------
Enter row (1-3): 3
Enter column (1-3): 3
```

```
-------------
Enter row (1-3): 3
Enter column (1-3): 3
-------------
| X |   |   |
-------------
| O | X |   |
-------------
|   | O | X |
-------------
X wins!
```