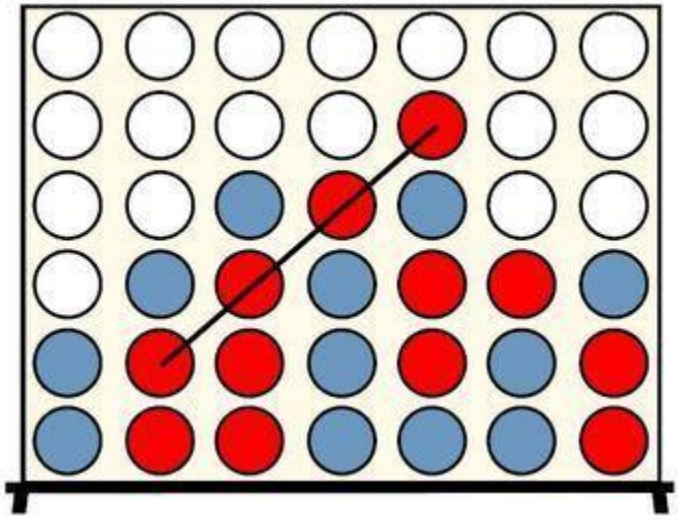
AI Lab Task: Implementing Connect-4 AI with Alpha-Beta Pruning

Connect 4 is a two-player strategy game played on a **6-row by 7-column grid**. Players take turns dropping discs into columns, and the goal is to be the first to form a horizontal, vertical, or diagonal line of four discs. The game ends when a player achieves this or when the board is full, resulting in a draw.



Objective:

In this lab, you will implement an AI for the game **Connect-4** using the **Minimax algorithm with Alpha-Beta Pruning**. Your implementation will involve:

1. Creating the Connect-4 board and generating valid moves.
2. Developing an evaluation function to assess board states.
3. Implementing the Minimax algorithm with Alpha-Beta Pruning to enable AI decision-making.
4. Simulating a game where a human player competes against the AI using the Minimax algorithm.

Instructions:

1. **Connect-4 Board Implementation**

Implement the following functions:

init_board() → Returns an empty board.

available_columns(board) → Lists all columns where a move can still be made.

display_board(board) → Prints the board to the console.

2. Score Function

Create a function **score_position(board, player)** that evaluates how favorable the board is for the given player.

Consider:

- **4-in-a-row:** High positive score for the player, negative for the opponent.
- **3-in-a-row and 2-in-a-row:** Moderate and small bonuses.
- **Blocking opponent's winning move:** Penalize the AI if it does not block an imminent win.

Make sure the function returns higher scores for better board states from the perspective of the current player.

3. Minimax Algorithm with Alpha-Beta Pruning

- Implement minimax(board, depth, alpha, beta, maximizing_player):
    - AI (Player 2) is the maximizing player, and the human (Player 1) is the minimizing player.
    - Implement Alpha-Beta Pruning:
        - alpha: Best value found for maximizing player.
        - beta: Best value found for minimizing player.
        - Prune branches that will not affect the final decision.
- Implement best_move(board) to return the best move for the AI.

4. Human vs. AI Game Simulation

- Implement play_game() where a human player plays against the AI.
- The human player chooses a column to drop a piece each turn.
- The AI responds using Minimax with Alpha-Beta Pruning.
- Limit the Minimax search depth to 4 for efficiency.

Constraints:

- Board size is **6x7**.
- Depth for Minimax is limited to **4** but you should test it with different depths.
- AI should calculate the best move using **Minimax with Alpha-Beta Pruning.**