Week2 - Lab Session

In this lab session, you will implement a board game Capture the flag. The objective of the game is to capture the flag ('F') before your opponent does. We have given some code for you to start your implementation. Read the game rules and complete the code.

Setup:
The game board is a square grid of a user-specified size.
We have given the code for you (char** initializeBoard(int size)).
Each player starts at a designated position on opposite corners of the board.
We have placed them
The flag is randomly placed on the board.
We have done it.

Gameplay:
Player 1's Turn:
Player 1 (represented by '1') starts at the top-left corner of the board (position [0, 0]).
Player 1 enters a move by specifying a direction (U/D/L/R for Up/Down/Left/Right).
Player 1's position on the board is updated according to the specified move.
If Player 1 lands on the flag position, Player 1 captures the flag and wins the game.
Player 2's Turn:
Player 2 (represented by '2') starts at the bottom-right corner of the board (position [size-1, size-1]).
Player 2 enters a move by specifying a direction (U/D/L/R for Up/Down/Left/Right).
Player 2's position on the board is updated according to the specified move.
If Player 2 lands on the flag position, Player 2 captures the flag and wins the game.

Game Continues:
Players take turns moving their characters until one of them captures the flag.
Players cannot move onto positions occupied by the opponent or out of bounds.
Game End:
The game ends when one of the players successfully captures the flag.
The player who captures the flag first wins the game.
The game board is displayed after each move, showing the current positions of both players and the flag.
Note:
Players should take turns inputting their moves.
The game can be replayed by running the program again. The position of the flag will be randomly generated each time.

Code
In labs folder, you will find the code. There are 4 different functions and the main function.

```
char** initializeBoard(int size);
void displayBoard(char** board, int size);
int hasCapturedFlag(char** board, int x, int y) ;
void movePlayer(char** board, int size, int* x, int* y, char direction);
```

1. char initializeBoard(int size);

We have already given the function.

Explanation:

- This function is responsible for creating and initializing a two-dimensional character array to represent the game board.
- It takes an integer parameter , which determines the dimensions of the square board.
- The function dynamically allocates memory for the board using and initializes each cell to a specific character, often representing an empty space or a default game piece.
- It returns a pointer to the allocated board.

2. void displayBoard(char **board, int size);

You should implement this function.

Explanation:

- This function displays the current state of the game board on the console.
- It takes two parameters:
    - char **: A pointer to the two-dimensional character array representing the game board.
    - int : The size of the board.
- The function iterates through the board and prints each character, creating a visual representation of the game state.

3. void movePlayer(char **board, int size, int *x, int* y, char direction);

You should implement this function.

Explanation:

- This function moves the player on the game board in the specified direction.
- It takes five parameters:
    - char **: A pointer to the two-dimensional character array representing the game board.
    - int : The size of the board.
    - int *: A pointer to the integer variable storing the x-coordinate of the player's position.
    - int *: A pointer to the integer variable storing the y-coordinate of the player's position.

  - char : A character representing the direction in which the player should move.
- The function checks if the move is valid by ensuring that the player stays within the boundaries of the board and does not collide.
- If the move is valid, the function updates the values of and accordingly to reflect the player's new position.

4. int hasCapturedFlag(char **board, int x, int y);

You should implement this function.

Explanation:

- This function checks whether the player at the specified position has captured the flag.
- It takes three parameters:
  - char **: A pointer to the two-dimensional character array representing the game board.
  - int: The x-coordinate of the player's position.
  - int: The y-coordinate of the player's position.
- The function checks the character at the given position on the board. If it matches the character representing the flag, the function returns 1; otherwise, it returns 0.