ASSIGNMENT-7

void move(int index, int gamer, int *flag):
- This function allows a specific player to take stones from a particular index and distribute them.
Parameters:
index: The index from which stones will be taken
gamer: Indicates which player will make the move (1 or 2)
*flag: Flag indicating whether the game is still ongoing
- The function takes stones from the chosen index and distributes them to other areas according to the rules. Then it updates the game board and checks the result.

void gamePlay(int gamer):
- This function allows a specific player to make a move.
Parameters:
gamer: Indicates which player will make the move (1 or 2)
-If the player is a human, it takes input from the user; if the player is a computer, it calculates the computer's move. Then it passes this move to the move function to update the game board.

int game_controller():
- This function checks whether the game has ended.
- Return Value: 1 (Game over) or 0 (Game still ongoing)
- By checking the number of stones each player has, it determines whether a player has run out of stones, indicating the end of the game.

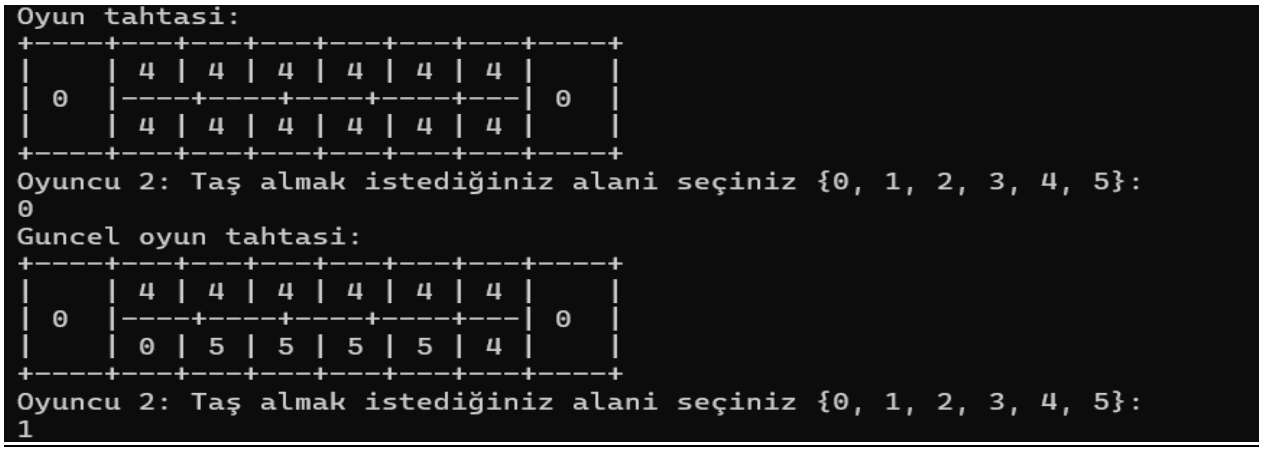
void initializeGame(int n):
- This function initializes the starting state of the game.
Parameters:
n: Determines the initial number of stones in each area
- It sets up the game board and initializes the number of stones each player has to start with.

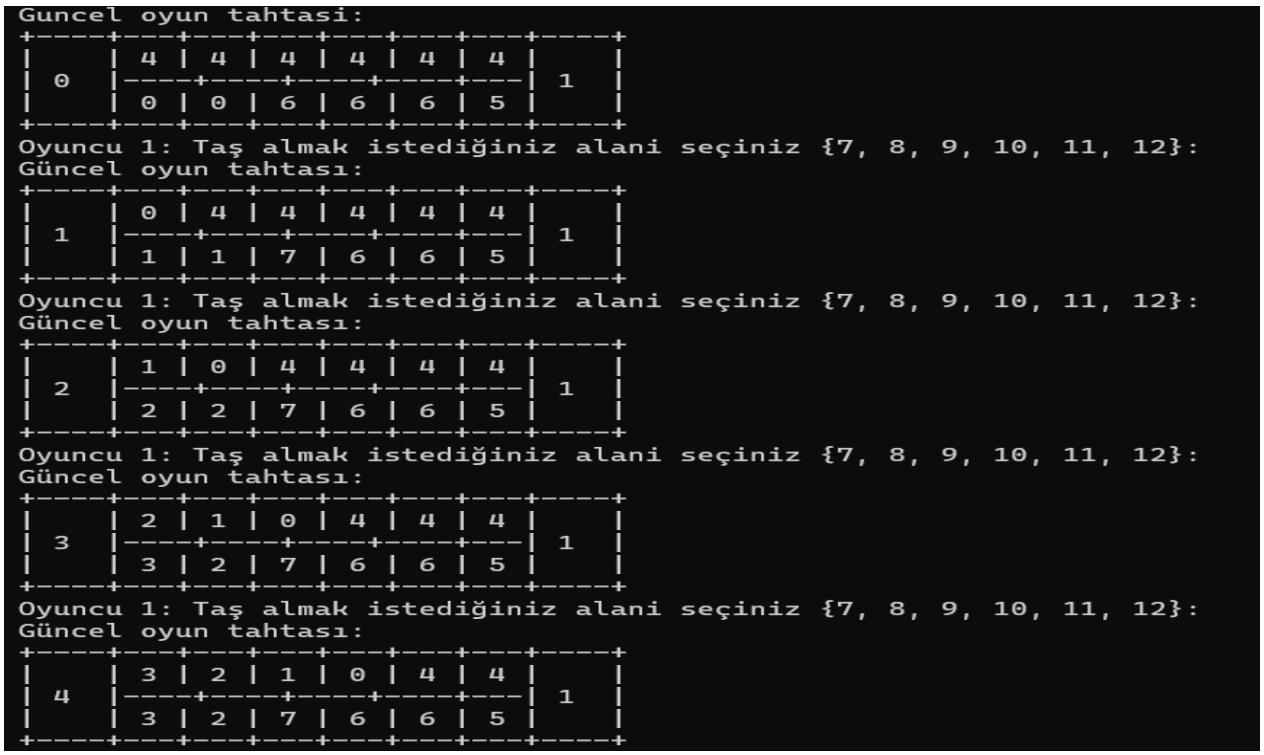
void print_board():
- This function prints the game board to the console.
It visually presents the game board to the user.

int computer(int pc_area[], int stone_counts[]):
- This function determines the computer player's move.
Parameters:
pc_area[]: Areas from which the computer player can make a move
stone_counts[]: Number of stones in each area
- It calculates which area to take stones from based on where it can collect the most stones.

- These functions manage different aspects of the game to control its flow and provide an interactive gaming experience to the users.

Example output :





Youtube Link :

https://youtu.be/bvLFIyp84EY