Details of submission

1. Design Principle- The idea was to create a 2048 game replica that runs on the output terminal. Priorities were

- ease of use,
- orderly display,
- functionality,
- the modularity of code,
- the readability of code
- efficiency

2. Problems faced- There was an issue with printing the code to the terminal efficiently however that was dealt with by having the string size of list members vary based on the largest member. Further, there was the issue of the output terminal not being readable over the process of the game, therefore a clear command was used.

3. Code walkthrough- The code consists of the following functions and a main which essentially transfers the inputs to the requisite functions. Functions are-

- show()- takes the linear array and displays it with blank spaces displayed as “_”
- shift()- given a 4x1 linear array, carries out a shift of all elements to the right with summing of the same elements
- rp(), lp(), up(), dp()- takes the linear array and uses shift appropriately to obtain new array
- newtile()- takes the linear array and adds a 2 in a random unoccupied location
- fox()-takes the linear array and converts it to a 2D array for purposes of display

The main code segment merely takes inputs repeatedly until there is no free tile or there is 2048 amidst the tiles.

4. This code can easily be ported to an 8x8 case by simply replacing the range(0,4) in every loop to (0,8) and replacing the parsed list from vertical pushes (up(), and dp()) to have 8 members in a vertical fashion.

5. This code can similarly easily endure a port to have 4096 as the required number by merely replacing the “if 2048 in ls” statement to “if 4096 in ls”. Although, the feasibility of users to solve the puzzle greatly decreases.