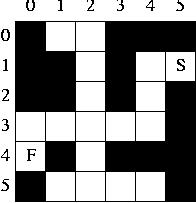
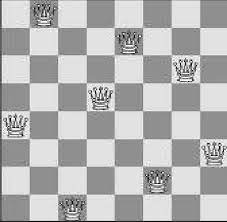
**Q1.** Write a recursive function to print the fibonacci sequence upto n terms.

**Q2.** A Maze is given as N\*N (generalise it for N) matrix of blocks where starting position is (1, N-1) and ending position is (N-2,0). You can move left, right, up and down on each cell. Blocked cells cannot be traversed and neither can you go out of bounds. Implement the recursive solution.

Take user input of value for N, then create a dynamic 2D array to store the maze. Initialise the contents of this maze randomly, with 0 for open cell or 1 for a blocked cell. Assign these values with 80% probability for an open cell (use rand function). Manually assign open cells for starting and end positions (these cannot be blocked). Then implement the algorithm for a recursive solution which will check whether the maze can be solved or not.

****

**Q3.** Write a recursive function to solve the n queens problem. You must arrange n queens in n rows on the chess board in such a way that no queen is under attack by another queen.



**Q4.** Write a recursive function to solve the n² /2 knights problem. You must arrange n² /2 knights in n rows on the chess board in such a way that no knight is under attack by another knight. For example, if n = 8, then 8² / 2 = 32. Meaning 32 knights can be arranged on the 8x8 chessboard without threatening each other Keep in mind that knights can attack in an L shape, as shown in the figure below:

