The Game of Life, also known simply as Life, was devised by the British mathematician John Horton Conway in 1970\textsuperscript{1}. The game operates on a two-dimensional grid of cells, each of which is in one of two possible states, alive or dead, or "populated" or "unpopulated". Every cell interacts with its eight neighbors, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. Any live cell with fewer than two live neighbors dies, as if caused by under population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The game keeps track of the status of cells in an array of \textit{bits} – one bit per cell. Your task is to implement the following two assembly language functions to store and retrieve individual bits within the array. You are to provide two solutions: The first version must be implemented without using bit-banding or loops. The second version must be implemented using bit-banding.

void PutBit(void *bits, uint32_t index, uint32_t bit);
uint32_t GetBit(void *bits, uint32_t index);

\textbf{Parameter Description}

\begin{tabular}{ll}
bits & Starting address where bits are stored in memory. \\
index & A bit position (0 to \#cells - 1) in the array of bits. \\
bit & Either 0 or 1. \\
\end{tabular}

Test your functions using the C main program that can be downloaded \url{here}. If your code is correct, the display should begin with a randomly distributed population, similar to the image at the right, but will of course change over time as cells are born (green) and die (red). Eventually the total population should stabilize. The blue button may be pressed to abort the game early. Once stabilized (or aborted), the screen will be erased and some statistics about the game will be displayed. Pressing the blue button again restarts the game.