

アルゴリズムとデータ構造 B 課題 2

二次元配列を用いてライフゲームを実装する。二次元配列の一つの要素をセルと捉え、隣接する 8 つのセルの状態により、次の世代のセルの状態（生 OR 死）が決まる。

問題 1

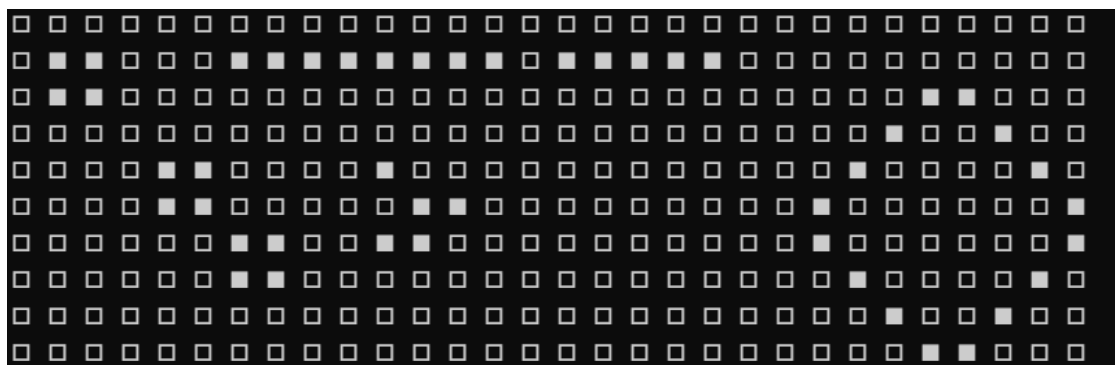
ライフゲームの状態を表現するための二次元配列を `malloc` 関数により動的に生成するプログラムを記述せよ。生成する二次元配列は `m` 行 `n` 列とし、`n`, `m` は `scanf` によって受け取る。配列の変数名は `life` とする。また、`life` は適切なタイミングで `free` 関数により解放すること。

問題 2

`input_kadai2.txt` から `life` に値を読み込むプログラムを記述せよ。

`input_kadai2.txt` には 10 行 30 列で 0, 1 の値を記入してある（※今回の課題では `n` として 30 を、`m` として 10 を入力することとする）。

さらに、ライフゲームの現在の状態を表示するプログラムを記述せよ。0 は□（死）で、1 は■（生）で表現すること。表示例は以下の通りである。



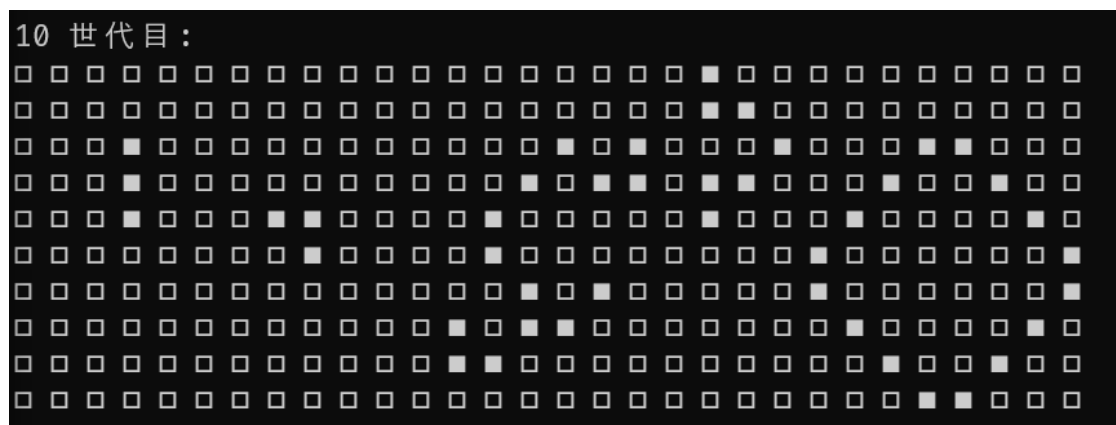
問題 3

隣接する 8 つのセルの状態に応じてセルの状態を更新するプログラムを記述せよ。隣接する 8 つのセルとは上下左右、斜めの 8 つを指す。更新のルールは以下の通りとする。

- 死んでいるセルに隣接する生きたセルがちょうど **3** つであれば、セルが生き返る
- 生きているセルに隣接する生きたセルが **2** つか **3** つであれば、セルは生き続ける
- 生きているセルに隣接する生きたセルが **1** つ以下であれば、セルは死ぬ
- 生きているセルに隣接する生きたセルが **4** つ以上であれば、セルは死ぬ

問題 4

世代数 **count** を **scanf** により受け取り、**count** の分だけライフゲームを実行（セルの状態を更新）し、各世代の状態を出力するプログラムを記述せよ（※今回の課題では世代数は **10** に固定する。つまり、**count** として **10** を入力する）。最終的な結果は以下の通りとなる。



課題の提出方法

問題 1~4 を一連のプログラムとして作成し、完成したプログラム全体（適切にコメントを付ける、スクリーンショット不可）、プログラムの実行結果（スクリーンショット）をレポートとしてまとめ、提出期限までに Teams で提出せよ。
 レポートのテンプレートファイル (word) は Teams に掲載してある。提出ファイル形式は PDF とする。

提出期限：2025 年 10 月 27 日（月）9:15