

課題 2

アルゴリズムとデータ構造B

第06回

演習 ex05.c の確認

課題 2

- レポート提出期限：2025年10月27日（月） 9:15

- 確保したい二次元配列の要素数は define で指定してある

```
#define N1 10 // 10人  
#define N2 5 // 5科目
```

✓ 本当は変数値で指定の方がよい

- malloc で N1行N2列の二次元配列を動的に生成

```
// 設問2 : 2次元配列に対応する2連鎖ポインタpを宣言する  
int **p;  
// 設問3 : 2次元配列を動的に生成  
p = (int**)malloc(N1 * sizeof(int*));  
for(i=0; i<N1; i++)  
    p[i] = (int*)malloc(N2 * sizeof(int));
```

- p[i], p の順に free

```
// 設問6 : 二次元配列領域の解放 (freeする順番に注意)  
for(i=0; i<N1; i++)  
    free(p[i]);  
free(p);
```

ex5.c 確認

- ファイルからの読み込みと書き込み

```
printf("設問4: input.txtから配列へデータを読み込み、確認のため出力\n");  
// 設問4: input.txtから配列へデータを読み込み、確認のため出力  
FILE *fp; // ファイルポインタを宣言  
fp = fopen("input.txt", "r"); // ファイルオープン  
for (i=0; i<N1; i++) {  
    for (j=0; j<N2; j++) {  
        // データの読み込み  
        fscanf(fp, "%d", &p[i][j]);  
        printf("p[%d][%d]:%d\n", i, j, p[i][j]); // 読み込みと同時に確認のため出力  
    }  
}  
fclose(fp);
```

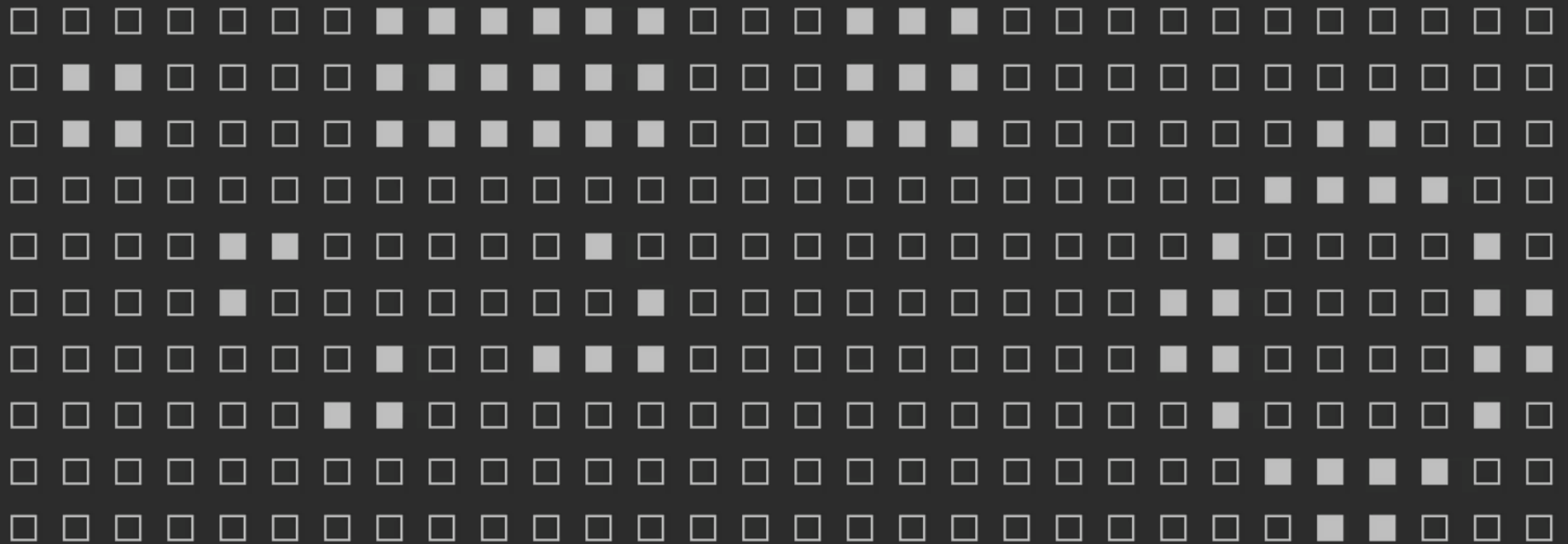
```
printf("設問5: 読み込んだデータ、合計点・平均点をoutput.csvに書き込み、確認のため合計点を出力\n");  
// 設問5: 設問5: 読み込んだデータ、合計点・平均点をoutput.csvに書き込み、確認のため合計点・平均点を出力  
fp = fopen("output.csv", "w"); // ファイルオープン  
fprintf(fp, "国語,数学,英語,理科,社会,合計,平均\n"); // csvのヘッダーを書き込み  
for(i=0; i<N1; i++) {  
    int total = 0;  
    float ave = 0;  
    for (j=0; j<N2; j++) {  
        // データの書き込み  
        fprintf(fp, "%d,", p[i][j]);  
        total += p[i][j]; // 合計点の計算  
    }  
    ave = (float)total / N2; // 平均点の計算  
    // 合計点の書き込み  
    fprintf(fp, "%d,", total);  
    // 平均点の書き込み  
    fprintf(fp, "%.2f\n", ave);  
    printf("p[%d] total=%d\n", i, total); // 確認  
    printf("p[%d] ave=%.2f\n", i, ave); // 確認  
}  
fclose(fp);
```

指示に従ってプログラムを作成し、
期限までにレポートを Teams 上で提出すること

- レポート提出期限：2025年10月27日（月）9:15

- ライフゲーム

1 世代目:



- m行n列の二次元配列の動的生成 ⇨ 第05回講義スライドと全く同じ
- input_kadai2.txt から life への値の読み込み ⇨ 演習 ex05.c とほぼ同じ
- 現在の状態の表示 ⇨ 二次元配列を走査し、値が 0 だったら□, 1だったら■
をif文で判定して表示

課題 2 「ライフゲーム」のキモは問題 3

● 方針（想定）：

- ✓ life とは別に更新後の状態を格納するための二次元配列（例えば new_life）を動的に確保
- ✓ life を走査しながら、隣接する8つのセルのうち生きたセルの数をカウント

```
for (int i = 0; i < m; i++) {  
    for (int j = 0; j < n; j++) {
```

- 現在のセルを i, j とすると、
int k が $i-1 \sim i+1$, int l が $j-1 \sim j+1$ のループで隣接8マス走査できる
 - ※ただし、 k が 0 より小さい、 m 以上、 l が 0 より小さい、 n 以上、
 k が i に等しくかつ l が j に等しいときは除く必要がある
(life の範囲外、もしくは、現在のセル自身のため)
- ✓ カウントした結果に応じて new_life の値を設定
 - ✓ life の走査が終わったら new_life の値で life の値を更新
 - ✓ new_life は適切に解放

$i-1, j-1$	$i-1, j$	$i-1, j+1$
$i, j-1$	i, j	$i, j+1$
$i+1, j-1$	$i+1, j$	$i+1, j+1$

- 初期状態と同じように，count 分状態を表示
 - ✓ スライド5みたいにアニメーションみたいにする必要は全くなく，普通に10個状態を表示してください