


```

//状態遷移関数
int trans_func(int tl, int tc, int tr,
               int ml, int mc, int mr,
               int bl, int bc, int br)
{
    //TODO!: 次の状態を決定する
    int N;
    N = tl + tc + tr + ml + mc + mr + bl + bc + br;

    if(N == 2 & mc == 1 | N == 3)
        return(1);
    else
        return(0);
}

#define XSIZE 10
#define YSIZE 10

int main()
{
    int i,t,x,y;

    //メモリ確保
    int* pmem = malloc(XSIZE*YSIZE*sizeof(int)*2);
    int* pcur = pmem;
    int* ppre = pmem + XSIZE*YSIZE;
    int* tmp;

    //pcur,ppreは1次元配列だが, x,y座標を基にインデクスを計算することで
    //2次元データを扱える

    // 例えば, XSIZE=8, YSIZE=3の2次元データは以下のように表現できる
    // (一桁の数字は座標, 2桁の数字はインデクス)
    //   0 1 2 3 4 5 6 7 (x)
    //   0 00 01 02 03 04 05 06 07
    //   1 08 09 10 11 12 13 14 15
    //   2 16 17 18 19 20 21 22 23
    //   (y)

    //TODO!: 初期状態設定 (一様な乱数など)
    int max = XSIZE*YSIZE;
    for(i = 0; i < max; ++i){
        int ic[i];
        ic[i] = rand() % 2;
        ppre[i] = ic[i];
    }

    for(t=0; t<3000; ++t) {
        // 端末のカーソルを左上に移動
        //これによりアニメーションが行える
        printf("%c[0;0H", 0x1b);

        for(y=0; y<YSIZE; ++y) {
            for(x=0; x<XSIZE; ++x) {
                //TODO!: 状態を文字として出力
                printf("%d ", ppre[y*XSIZE+x]);

                //TODO!: セルに状態遷移関数を適用
                //x,y座標からアドレスを計算し参照, 代入する.
                //端に関しては, 特別な処理が必要. 両端はつながっているものとする
                if(x == 0 & y == 0)
                    pcur[y*XSIZE+x] = trans_func(ppre[XSIZE*(YSIZE)-1], ppre[(XSIZE)*(YSIZE)-1]),

```

```

ppre[(XSIZE)*(YSIZE-1)+1],
ppre[y*XSIZE+x+1],
ppre[y*XSIZE+x+(XSIZE+1)];

    else if(x > 0 & x < XSIZE-1 & y == 0)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x+(XSIZE*(YSIZE-1)-1)],
ppre[y*XSIZE+x+(XSIZE*(YSIZE-1))], ppre[y*XSIZE+x+(XSIZE*(YSIZE-1)+1)],
ppre[y*XSIZE+x-1], ppre[y*XSIZE+x],
ppre[y*XSIZE+x+1],
ppre[y*XSIZE+x+(XSIZE-1)], ppre[y*XSIZE+x+(XSIZE)],
ppre[y*XSIZE+x+(XSIZE+1)]);

    else if(x == XSIZE-1 & y == 0)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x+(XSIZE*(YSIZE-1)-1)],
ppre[y*XSIZE+x+(XSIZE*(YSIZE-1))], ppre[XSIZE*(YSIZE-1)],
ppre[y*XSIZE+x-1], ppre[y*XSIZE+x],
ppre[y*XSIZE+x-(XSIZE-1)],
ppre[y*XSIZE+x+(XSIZE-1)], ppre[y*XSIZE+x+(XSIZE)],
ppre[y*XSIZE+x+1]);

    else if(x == 0 & y > 0 & y < YSIZE-1)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x-1], ppre[y*XSIZE+x-(XSIZE)],
ppre[y*XSIZE+x-(XSIZE-1)],
ppre[y*XSIZE+x+(XSIZE-1)], ppre[y*XSIZE+x],
ppre[y*XSIZE+x+1],
ppre[y*XSIZE+x+(XSIZE*(YSIZE-1)-1)], ppre[y*XSIZE+x+(XSIZE)], ppre[y*XSIZE+x+(XSIZE+1)]);

    else if(x == 0 & y == YSIZE-1)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x-1], ppre[y*XSIZE+x-(XSIZE)],
ppre[y*XSIZE+x-(XSIZE-1)],
ppre[y*XSIZE+x+(XSIZE-1)], ppre[y*XSIZE+x],
ppre[y*XSIZE+x+1],
ppre[XSIZE-1], ppre[0], ppre[1]);

    else if(x > 0 & x < XSIZE-1 & y == YSIZE-1)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x-(XSIZE+1)], ppre[y*XSIZE+x-(XSIZE)],
ppre[y*XSIZE+x-(XSIZE-1)],
ppre[y*XSIZE+x-1], ppre[y*XSIZE+x],
ppre[y*XSIZE+x+1],
ppre[y*XSIZE+x-(XSIZE*(YSIZE-1))-1], ppre[y*XSIZE+x-(XSIZE*(YSIZE-1))],
ppre[y*XSIZE+x-(XSIZE*(YSIZE-1))+1]);

    else if(x == XSIZE-1 & y == YSIZE-1)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x-(XSIZE+1)], ppre[y*XSIZE+x-(XSIZE)],
ppre[y*XSIZE+x-(XSIZE*2-1)],
ppre[y*XSIZE+x-1], ppre[y*XSIZE+x],
ppre[y*XSIZE+x-(XSIZE-1)],
ppre[y*XSIZE+x-(XSIZE*(YSIZE-1))-1], ppre[y*XSIZE+x-(XSIZE*(YSIZE-1))], ppre[0]);

    else if(x == XSIZE-1 & y > 0 & y < YSIZE-1)
        pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x-(XSIZE+1)], ppre[y*XSIZE+x-(XSIZE)],
ppre[y*XSIZE+x-(XSIZE*2-1)],
ppre[y*XSIZE+x-1], ppre[y*XSIZE+x],
ppre[y*XSIZE+x-(XSIZE-1)],
ppre[y*XSIZE+x+(XSIZE-1)], ppre[y*XSIZE+x+(XSIZE)],
ppre[y*XSIZE+x+1]);

    else pcur[y*XSIZE+x] = trans_func(ppre[y*XSIZE+x-(XSIZE+1)], ppre[y*XSIZE+x-(XSIZE)],
ppre[y*XSIZE+x-(XSIZE-1)],

```

```

ppre[y*XSIZE+x+1],
ppre[y*XSIZE+x+(XSIZE+1)];
    }
    putchar('\n');
}
//!!!!ポインタを交換
ptmp = pcur;
pcur = ppre;
ppre = ptmp;
//ウエイト
//アニメーションの速度を調節する
sleep(1);
}

free(pmem);
}

```

(2)実行結果

(途中経過)

```

0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 1 0 0 0

```

(最終状態)

```

0 0 0 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

(3)近傍則

近傍8セル（着目セルに接する全てのセル）の状態の合計値をNをして

- ・ N=2で且つ着目セルの状態（すなわち状態i）が1のときは、次状態は1
- ・ N=3の時、次状態は1
- ・ それ以外の時、次状態は0

という近傍則を用いた