

生命現象や神経発火現象を再現する離散的計算モデルに、1次元セルオートマトンがある。1次元に並んだマス目をセルと呼び、各セルは状態0または1をとる。あるセルとその両隣のセルを合わせた3近傍の状態を000のように表現する。時間は離散的に進み、時刻 t から $t+1$ への3近傍の変化は、3近傍の状態とあらかじめ決められたルールにより決まる。ルールはルール番号毎に異なる。ルール番号は3近傍の全状態<111、110、101、100、011、010、001、000>に対して、各3近傍の中央セルの時刻変化後の値を2進数とし、それを10進数に変換し求められる。

例えば、3近傍の全状態の変化が以下のように定められた時を考える。

3近傍セルパターン (時刻 t)	<111	110	101	100	011	010	001	000>
3近傍の中央セル (時刻 $t+1$)	0	1	0	1	1	0	1	0

この時のルール番号は、時刻 $t+1$ のパターン01011010を2進数表現 $(01011010)_2$ と見て、これを10進数表現90に変換し、ルール番号90と定義される。

このルール番号90に従えば、例えば時刻 t で001001101010110であった1次元セルは、時刻 $t+1$ では次のように変化する。但し両端は常にダミーとして0とする。

時刻 t	001001101010110
時刻 $t+1$	010111100000110

また次ページには、ルール番号 n を入力すると、 $LEN+1$ 個の1次元セルの時刻変化を出力するC言語プログラムの一部を示した。1次元セルオートマトン及び次ページのプログラムについて以下の問いに答えよ。

問1：ルール番号の最大値はいくつになるか、解答用紙に記せ。

問2：次ページのプログラムに、ルール番号110を入力した場合、(ア)の時点での配列 c の値はいくつになるか。 $c[0]$ から $c[7]$ まで解答用紙に記せ。

問3：時刻 $t=0$ の時、1次元セルの状態が0000000010000000だったとする。ルール番号30に従う場合、この1次元セルの状態はどのように時刻変化するか、時刻 $t=1$ から10まで解答用紙に記せ。

問4：下記のような1次元セルの時刻変化が観察された時、ルール番号は何番であるか？考えられるルール番号を全て解答用紙に記載せよ。

時刻 t	001000110001010
時刻 $t+1$	001000110001010
時刻 $t+2$	001000110001010

問5：プログラム中の空欄(1)～(2)に入る答えを解答用紙に記せ。

【プログラム】

```
#include <stdio.h>
```

```
#define LEN 14
```

```
//ルール番号を入力する
```

```
void rule_set(int c[]){
```

```
    int n;
```

```
    printf("ルール番号の入力:");
```

```
    scanf("%d", &n);
```

```
//ルールを求める
```

```
for(int i=0; i<8; i++){
```

```
    (1)
```

```
}
```

```
}
```

```
//ルールを適用し3近傍の中央セルの状態を求める
```

```
int rule_apply(int c[], const int left, const int center, const int right){
```

```
    (2)
```

```
    const int next = c[n];
```

```
    return next;
```

```
}
```

```
void init(int a[]){
```

```
    //セルの初期値を決める
```

```
}
```

```
void show(int a[]){
```

```
    //セルの中身の0か1を出力する。
```

```
}
```

(次ページに続く)

```
// (セルを) コピー
void copy(int dst[], int src[]){
    //src を dst にコピーする
}

int main(void){
    int a[LEN+1], b[LEN+1];
    int c[8];
    const int time_max = 40;

    init(a);

    rule_set(c);
    ←----- (ア)
    //時刻 t、t+1 に来る 1 次元セルを決める
    for(int t=0; t<time_max; t++){
        //時刻 t での出力
        show(a);

        //a を b にコピー
        copy(b, a);

        //時刻 t+1 の 1 次元セルの両端を 0 にする
        a[0] = 0;
        a[LEN] = 0;

        //時刻 t+1 の端を除いたセルの状態を決める
        for(int s=1; s<=LEN-1; s++){
            a[s] = rule_apply(c, b[s-1], b[s], b[s+1] );
        }
    }
    return 0;
}
```