

アルゴリズムとデータ構造1 演習問題 (1) 回答

C 言語との主たる違いは、下記である。

1. 変数への代入文を = ではなく、:= を用いて表していること。
2. ポインタを示す * の代わりに、↑ を用いていること。

```
main ( )
/* 1~6 の整数の一樣乱数と, [0,1] の範囲の実数の一樣乱数を 1000 個生成し, 配列 A[·] に入れる */
/* 混合合同法  $x[i+1] = (a \cdot x[i] + b) \bmod c$  を用いて生成する. 初期値  $x[0]$  は入力する */
/* ここでは,  $a = 17, b = 5, c = 1024$  を定数として用いる */
{
    typedef struct {
        int     dice; /* 欄 dice は 1~6 の整数を持つ */
        float   rand; /* 欄 rand は [0,1] の範囲の実数を持つ */
    } RCELL;
    maxlength = 1000;
    a = 17;
    b = 5;
    c = 1024;

    int i; /* 1 ~ maxlength の繰り返しに用いる制御変数 */
    RCELL A[1..maxlength]; /* 配列 A[·] の各要素は RCELL 型の構造体 */
    int x; /* 0 ~ c-1 の一樣乱数を生成するため変数 */

    printf( “正整数を入力せよ \n” ); /* 正整数を一つ入力せよと指示を出す */
    getint( &x ); /* 入力された正整数を x とする */
    x := x % c; /*  $x := x \bmod c$ ; */
                /* 読み込んだ正整数を c で割った余りを x とする */

    for( 1 i maxlength なる各 i. i の昇順に ){
        x := GenerateRand( x, a, b, c, &A[i].rand ); /* x を更新し, A[i] の rand も計算する */
        A[i].dice := ( x % 6 ) + 1; /* A[i] の欄 dice に 1~6 の整数を入れる */
                                   /*  $x \% 6$  は  $x \bmod 6$  で, x を 6 で割った余り */
    }

    A[·] の各値を出力する ;
} /* main */
```

```
int GenerateRand( int x, int a, int b, int c, float↑ prand )
/* 与えられた x から, 混合合同法  $x := (a \cdot x[i] + b) \bmod c$  を用いて新たな乱数 x を生成する */
/* [0,1] の範囲の実数 rand も生成し, prand が指す領域に入れる */

{
    x := ( a*x + b ) % c; /*  $(a*x + b) \% c = (a*x + b) \bmod c$  */
    ↑prand := x / ( c - 1 ); /* prand が指す領域に実数  $x / (c - 1)$  を入れる */
    return x; /* 新たに生成した x を返す */
} /* GenerateRand */
```