第4問

コンピュータのプログラムで発生できる乱数は，そのプログラムの処理に従って発生するということから，真の乱数ではなく，より正確には擬似乱数と呼ぶべきものである．しかし，以下では，擬似乱数を単に乱数と呼ぶことにする．このような乱数に関し，次の問いに答えよ．なお，プログラムを解答する場合，C，Java，LISP，Pascal などの一般的に知られたプログラミング言語を用いて記述せよ．また，解答に用いたプログラミング言語の名称を明記せよ．

(1) コンピュータのプログラムで発生させる乱数が真の乱数にできる限り近くなるための3条件を，等確率，自己相関，周期という用語をそれぞれ用いて具体的に説明せよ．

(2) 0から $m-1$（m は $m>1$ を満たす整数）までの範囲の整数を乱数（$x_1, x_2, \cdots$）として発生することを考える．各乱数 x_j（$j=2, 3, \cdots$）は，x_1 を初期値とし，a と c を整数（$a>0$，$c \geqq 0$）として，$(a \cdot x_{j-1} + c)$ を m で割った余り，つまり，

$$x_j = (a \cdot x_{j-1} + c) \bmod m,$$

として順次導出する．

この方法で乱数を n 個発生し出力するプログラムを記述せよ．ただし，乗算（整数 a と直前の乱数の掛け算）におけるあふれは考慮しなくてよい．

(3) (2)の乱数発生法において乗算であふれが生じると，好ましくない乱数が発生することがある．$c=0$ の場合，乗算におけるあふれを回避するため，次式によって x_j を計算する方法がある．

$$x_j = \begin{cases} a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r & \cdots a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r \geq 0 \\ a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r + m & \cdots a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r < 0 \end{cases}$$

ここで，$\{z/w\}$ は z を w で割った商（負でない整数）を表すものとし，$q = \{m/a\}$，$r = m \bmod a$，即ち $m = a \cdot q + r$ で，$r<q$ とする．

この方法における乗算ではあふれが生じないことを証明せよ．

(4) もう一つの乱数発生法として，図1に回路を示す線形フィードバックシフトレジスタを用いる方法がある．図1の回路では，容量が1ビットの各レジスタ y_i（i=1, 2, 3, 4）に予め初期値を設定しておき，1ステップ動作毎，右にシフトすると同時に，右端の2個のレジスタ y_3 と y_4 の値の排他的論理和演算によって得られる値を乱数として出力するとともに，最も左のレジスタ y_1 に入力する．

各レジスタの初期値をすべて 1 とした場合，ステップ動作を繰り返すことによって得られる全レジスタの値を順次求めよ．

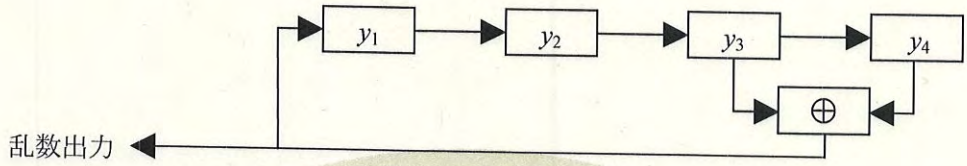

図 1

(5) (4)の乱数発生法において，各レジスタが保持する値を整数に変更し，排他的論理和演算を整数の加算に変更する．この変更後の乱数発生法で，レジスタの個数を k（k>2）として，各レジスタを y_i（$i = 1, 2, \cdots, k$）とし，各ステップで加算の対象となるレジスタを y_j と y_k とする（$1 \leq j < k$）．

この乱数発生法に従って，0 から m−1（m は m >1 を満たす整数）までの範囲の整数の乱数を，呼ばれるたびに 1 個発生して返す関数のプログラムを記述せよ．ただし，y_i（$i = 1, 2, \cdots, k$）に対する演算の実行回数をできる限り少なくし，大域変数の使用はできる限り少なくせよ．なお，y_i の初期値は既に設定済みであると仮定してよい．

Problem 4

Random numbers that can be generated by a computer program are not truly random because they are generated in accordance with the processing of the program and thus these numbers should be called pseudo random numbers to put it more precisely. However, these pseudo random numbers are called random numbers briefly hereinafter. Answer the following questions with regard to such random numbers. When writing a program, use one of well-known programming languages such as C, Java, LISP and Pascal, and identify the name of the programming language used in the answer.

(1) Answer three conditions to be satisfied by random numbers generated by a computer program so that these numbers are as close to true random numbers as possible. Here, these three conditions should be described concretely using each of the following technical terms: equal probability, auto-correlation and period.

(2) Consider generating random numbers (x_1, x_2, $\cdots$) in the range from 0 to $m-1$, where m is an integer that is larger than 1. Each random number x_j ($j = 2, 3, \cdots$) is sequentially derived as the remainder after the division of $(a \cdot x_{j-1} + c)$ by m, that is,

$$x_j = (a \cdot x_{j-1} + c) \bmod m,$$

where x_1 is the initial value, and a and c are integers ($a>0$, $c \geq 0$).

Write a program that generates and outputs n random numbers according to this method. Here, an overflow at the multiplication (of integer a and just preceding random number) can be ignored.

(3) If there is an overflow at the multiplication in the random number generation method of (2), non-preferable random numbers may be generated. A method to calculate x_j avoiding overflows when $c = 0$ is to use the following formulae.

$$x_j = \begin{cases} a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r & \cdots a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r \geq 0 \\ a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r + m & \cdots a \cdot (x_{j-1} \bmod q) - \{x_{j-1}/q\} \cdot r < 0, \end{cases}$$

where, $\{z/w\}$ denotes the (non-negative integer) quotient modulo w of z, $q = \{m/a\}$, $r = m \bmod a$, that is, $m = a \cdot q + r$, and $r < q$.

Prove that there is no overflow at the multiplications in this method.

(4) Another method for generating random numbers makes use of a linear feedback shift register whose circuit is shown in Fig. 1. In the circuit of Fig. 1, the initial values are preset at registers y_i (i=1, 2, 3, 4), each of which has a capacity of 1 bit. The values of the registers are shifted toward right, and the exclusive OR operation result of the rightmost two registers y_3 and y_4 becomes an output of a random number and is provided to the leftmost register y_1 in accordance

with each step operation.

Show the sequence of values of all the registers obtained by repeating the step operations assuming that the initial values of the registers are all 1s.

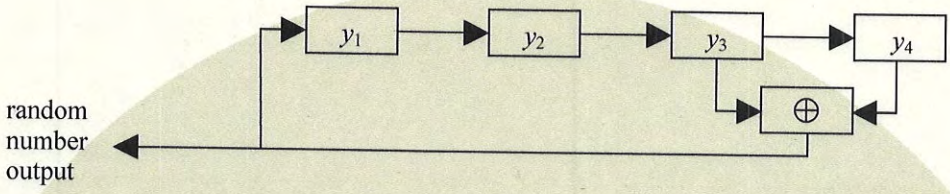

Fig. 1

(5) The value of each register is changed to an integer and the exclusive OR operation is replaced by the addition of integers in the random number generation method of (4). Let y_i (i=1, 2, ⋯, k) denote each register, where k (k >2) is the number of registers and let y_j ($1 \leq j < k$) and y_k denote two registers whose values are to be added in the random number generation method derived by these changes.

Write a program of a function that generates and returns a random number in the range from 0 to m−1, where m is an integer that is larger than 1, according to this new random number generation method every time the function is called. Here, the number of executions of operations on y_i (i=1, 2, ⋯, k) and the number of global variables used should be made as small as possible. The initial values of y_i are all assumed to be preset.