計算機数学 NO.14

第 14 回目の主題：**ruby でプログラミング (2)**

1. 今日すること

ruby でプログラミングを行ない、正の数 a, b, m と (大きな) 数 n に対して、a^n を m で割ったあまりを計算する関数 $f(a, n, m)$ を作る。実際に実行してみてプログラムとその結果を verbatim を用いて TeX に取り込んで提出せよ。

2. ヒントと問題

◎アイディアその 1. $f(a, n, m) = a^n \mod m$ を求めるのに、わざと少し余分な c も考えて、$g(c, a, n, m) = ca^n \mod m$ を作る。($f(a, n, m) = g(1, a, n, m)$ である。)
◎アイディアその 2. べき指数 n を 2 でわって、$n = 2q + r$ と書くと

$$ca^n \mod m = (c \cdot a^r)(a^2)^q \mod m$$

◎レベル 5.

```
def torikae(c,a,n)
  r=n% 2                ### r は c を 2で割った余り
  if (r== 0 )           ### 条件文の ruby 的書き方。
    c1=c
  else
    c1=c*a
  end
  a1=a**2
  n1=n/2                ### n1=n.div(2) のほうがいいかも。(付記参照)
  return([c1,a1,n1])
end
```

上のように (正しく) 入力した後、`torikae(1,2,5)` を実行すると、何が得られるか?
◎利用例 (レベル 6)(レベル 5 の続きに書く。)

```
def g1(c,a,n)               ##  c*(a**n) を求める関数
  while n>=1                ## 「n>=1 のあいだ繰り返す」の ruby 的書き方。
    c,a,n=torikae(c,a,n)    ## このように変数をいっぺんに代入できる。
  end
  return(c)
end
```

○一般に正の数 c, a, n を レベル 5 の torikae(c,a,n) の出力結果で取り替えちゃった後でも ca^n の値は変わらないことを納得せよ。(納得するだけでいい。)
◎最終問題 (レベル 7): $n = 10^{10} + 19$ のとき、$2^{(n-1)/2}$ を n で割ったあまりを求めよ。
ヒント: `torikae(c,a,n)` の計算の後、c, a を それぞれ $c\%m$, $a\%m$ で置き換えるようにして本日の課題の f, g をつくり、計算を実行する。
◎ f の動作確認プログラム例

```
for i in 1..10
p [i,f(2,i,10**10)]
```

○付記:
ruby では、100/3 は 100 を「あまりを許した割り算」で 3 で割った商 (つまり 33) を指すのが標準の動作である。しかし場合によっては (プログラム側で動作を指定することにより) 10/3 を分数 (33.333... と等しいアレ) と認識する場合もある。そのような場合、上で 10/3, n/2 などとある部分は、それぞれ `10.div(3)`, `n.div(2)` などと書いてやるとよい。