

1.2 10 進数と 256 進数

[Fibonacci.vba] で試して分かったと思うが、フィボナッチ数列は結構大きな数になってしまふ。君たちは、何項目まで求めることができただろうか。コンピュータのことだから、無限に大きな値を扱うわけにはいかないが、やりようによっては数万桁の数でも扱える。しかし、この話題はしばらく後になる。

フィボナッチ数の計算に使った変数は Integer 型の整数だったはずだ。コンピュータが扱える数値に限界があるのは、変数の型による制約が大きい。われわれが日頃使っている数は 10 進数である。これは 1 桁あたり 10 種類の数を使える。携帯電話は 090-xxxx-xxxx の番号を持っているので、自由になる桁は 8 桁ある。各桁に 10 種類の数が使えるから、電話番号は

$$10 \times 10 = 10^8 = 1 \text{ 億}$$

まで OK だ。残念ながら、国民 1 人 1 人には行き渡らない計算だが。

さて、VBA の“桁”にあたる単位は“バイト”という。1 バイトは xxxxxxxx の 2 進数で処理されている。具体的には 11010010 や 01011011 などという種類の“数字”が使えるのだ。すると 1 バイトで $2^8 = 256$ 種類の“数字”になる。従って、VBA は 256 進数ということなのだ。何だか目が回りそうになってないかい？ 気を確かに持とう。

そこでようやく Integer 型の整数の話に入る。Integer 型の整数は常に 2 バイトである。2 バイトなら $256^2 = 65536$ 通りの数が使える。しかし Integer 型は整数を前提としている—つまり半分が正の数に、残りの半分が負の数に使われる—ので、2 バイトなら -32768 ~ 32767 の整数が扱えるのだ。

大きな数を扱いたければ桁数を増やせばよいのだから、バイト値を増やせばいくらでも大きな桁を扱えることになる。もっとも、無制限に大きな桁を扱えるほど、型が用意されているわけではないのだが。Integer 型より大きい桁を扱いたいときは Long 型を使えばよい。これは 4 バイトの大きさの整数だ。

programming list [Fibonac2.vba]

```

1 : Sub Fibonac2()
2 : Const SUP As Integer = 20
3 : Dim f0, f1 As Long
4 : Dim i As Integer
5 :
6 :     f0 = 1: f1 = 1
7 :     For i = 3 To SUP
8 :         Sheet1.Cells(i, 1) = f1 + f0
9 :         f1 = f1 + f0
10 :        f0 = f1 - f0
11 :    Next i

```

12 : End Sub

今回のマクロは [Fibonacci.vba] と較べて、f1, f0 を Long 型にしたことだけである。i は Integer 型のままだ。大は小を兼ねるとは言え、プログラミングに関しては無駄でしかない。すべてを Long 型にして手間を省こうとしないで、常に最適な型を使うよう心掛けたいね。さて、こうすることで、何番目のフィボナッチ数まで計算できるだろうか？

ここでコンピュータが内部で処理する数について触れておこう。今、256 進数などと大袈裟なことを書いたが、コンピュータは内部では 2 進数で計算している。2 進数は 0 と 1 の 2 種類の数字で様々な数を表現する。だから 2 進数には、101 とか 11100 とか 10110111 とかの表現しかできない。これらは 10 進数では、それぞれ 5, 28, 183 を表している。むむ。では、正の数と負の数の区別はどこでするんだろう？

さっき、1 バイトが XXXXXXXX の 2 進数で処理されると言ったのを覚えているだろうか。この場合、コンピュータは 8 桁すべてに数字を埋めるのだ。10 進数を例にとると、われわれなら 12345 と書く数でもコンピュータは 00012345 とする。これがミソだ。

そこでコンピュータは、最上位の桁によって正負の判断をすることが可能になる。実際は、最上位が 0 なら正の数、1 なら負の数と決めている。しかし間違えないでほしい。1 は負の数であることを示しているのであって、決して「マイナスの記号の代わりではない」ということなのだ。このことについては、しばらく後で、別のマクロに関連して話すことにしよう。