

## 6...の旅

### 6.1 完全数

6 は興味深い数のひとつだ。それはどういうことかという、6 の約数は 1, 2, 3, 6 だが、

$$1 + 2 + 3 = 6$$

となっている。この性質を満たす数は 6 の近辺にはない。例えば 8 は、約数として 1, 2, 4, 8 を持つが、明らかに  $1 + 2 + 4 \neq 8$  である。約数をもれなく取り出すことができれば、6 の次にこの性質を満たす数が 28 であることを知るの容易い。実際、28 の約数は 1, 2, 4, 7, 14, 28 であるから

$$1 + 2 + 4 + 7 + 14 = 28$$

になっている。この性質を持つ数を完全数と呼ぶ。

性質を誤解のないように言えば、完全数とは

自分自身を除く約数の総和が自分自身に等しい数

であると言える。なんと、日本語にするほうが式で説明するより難しくなっていないかい？ 数学とはそんなものだ。

それなら、28 の次の完全数はいくつだろうか。ちょっと計算すればすぐに見つかると思うだろう。悪いね。そんな簡単には見つからないのだ。よほど根気よくなければ、次の完全数を見つけるのは難しい。根気が勝負になるときこそコンピュータの出番というわけだ。

いきなり完全数を求めるマクロを組むのも大変だろうから、まずは約数を見つけるマクロから見ていこう。

---

programming list [DispOfCM.vba]

```
1: Sub DispOfCM()  
2: Dim n, i, col As Integer  
3:  
4:     n = Sheet1.Cells(1, 1).Value  
5:     col = 1: Sheet1.Cells(2, col) = 1  
6:
```

```

7:      For i = 2 To n / 2
8:          If n Mod i = 0 Then
9:              col = col + 1: Sheet1.Cells(2, col) = i
10:         End If
11:     Next i
12:
13:     col = col + 1: Sheet1.Cells(2, col) = n
14: End Sub

```

ここでは新たな命令は一切登場しない。出力の一部にほんのり違いがあるだけだ。

プログラムは A1 セルに入力されている数に対して、約数をすべて表示することになってる。ところである数が  $n$  のとき、約数に 1 と  $n$  があるのは当然なので、これらは最初と最後に表示することにした。ここが、ほんのり違うところである。

まず、4: 行目で約数を調査される数が取り込まれる。5: 行目は明らかな最初の約数 1 の表示である。

7: 行目からの For-Next 構文は、 $n$  を 2 から順々に割っていくものだ。約数を求める行為は素数を求める行為と違うので、2 で割れても 4 で割れるか試す必要があるし、1 ずつ大きな数で割る必要もある。ただし、 $n / 2$  までの数で試せばよい。それより大きい数が約数になることはないのだから。

8: 行目は  $n$  が  $i$  で割れるかどうかの確認だ。割れれば約数であるから 9: 行目に表示し、割れなければ何もせず次の  $i$  を試すのだ。For-Next 構文で約数を見つけたら、13: 行目で最後の約数——つまり自分自身——を表示すれば終了だ。

また、マクロは約数を列挙するものだが、素数を見つける役にも立つ。なぜなら、素数は 1 と自分自身しか約数を持たない数だから、例えば 17 に対しては 1 と 17 だけが A2 セルと B2 セルに表示される。

これで、大きな数の約数を調べることができるようになった。例えば自分の誕生日を西暦年からつなげると大きな数になる。1995 年 4 月 19 日生まれなら 19950419 だ。さて君たちの誕生日はどれくらい約数を持つだろうか。それとも素数だろうか。

ところで、ここに提示した約数を求めるマクロは、単純きわまりないアルゴリズムであるだけに無駄が多い。当然のことながら、例えばある数  $n$  が  $i$  で割れたなら、その商である  $\frac{n}{i}$  も約数である。つまり約数は基本的に 2 つ同時にみつけることができる。基本的と表現したのは、49 が 7 で割れたからと言って、 $\frac{49}{7}$  が別の約数とは言えないからだ。しかし、除数  $i$  と商  $\frac{n}{i}$  を一緒に表示してしまえば、割り算は  $\sqrt{n}$  まで試せばよいことになる。これは以前、素数を求めた際に使った手法だし、計算量を減らす効果も十分だ。余裕があればこの考え方でマクロを書き換えてほしい。

だが、完全数を求めるためには、これではまったく不十分なのだ。後で分かるように、完全数は素数とは較べものにならないくらい稀にしか現れない。つまり、計算量が減ること自体は喜ばしい

けれど、結局ほとんどの数で調査が不発に終わる。効果的に調べるには、計算量を減らすのではなく、調査しなくてもよい数を飛ばすことである。