0.5 循環小数の秘密

もうしばらく小数の話題を続けてみよう。

小数には $0.333\cdots$ のような**無限小数**と、0.125 のような**有限小数**がある。また、ひと口に無限小数といっても、$0.333\cdots$ は**循環小数**と呼ばれる数で、円周率 $3.141592\cdots$ のように**循環しない小数**とは区別している。実は、すべての有理数は有限小数か循環小数になる。言い換えれば、分数は必ず有限小数か循環小数にできるということで、決して循環しない無限小数にはならない。その逆に、循環しない無限小数は決して分数にすることはできない。どういうことだろうか。

例えば $\dfrac{1}{6}$ は循環する無限小数である。実際に割り算を行ってみれば一目瞭然だろう。

$$
\begin{array}{rr}
 & 0.\ 1\ 6\ 6 \\ \cline{2-2}
6\ \big) & 1.\ 0\ 0\ 0 \\
 & 6 \\ \cline{2-2}
 & 4\ 0 \\
 & 3\ 6 \\ \cline{2-2}
 & 4\ 0
\end{array}
$$

割り算は途中で計算を止めているが、このあとは 6 が続くだけである。その理由は簡単だ。割り算の最後の行に余りである 40 がある。そしてこれと同じ余りがひとつ前にも出ているね。そう、小数が循環する理由は、以前の余りと同じものがでるからなのだ。

分数を小数に直すときは、分子を分母で割るはずである。そのときの余りは、割る数である分母より小さい数しかありえない。具体的には、6 で割り算をすれば余りは 0, 1, 2, 3, 4, 5 の 6 種類に限られる。つまり余りの種類は、0 を含めて**高々**分母に使われた数だけしかないのだ。そのせいで余りにあたる数は、いつか必ず同じものになってしまう。一度同じ余りになれば、あとは循環するしかないし、余りが 0 になれば割り切れるということなのだから。

それでは、循環する無限小数は、始めどんな分数だったか気にならないだろうか？　しかし、循環する無限小数をもとの分数に復元するのは簡単である。$0.1666\cdots$ であれば $x = 0.1666\cdots$ とおいて

$$
\begin{array}{rrcl}
 & 100x & = & 16.666\cdots \\
-) & 10x & = & 1.666\cdots \\ \hline
 & 90x & = & 15
\end{array}
$$

のようにすれば、$x = \dfrac{15}{90}$ であることが分かるのだ。この方法はどんな循環小数にも使える。コツは循環する部分がそろうように、**適当な** 10 の倍数を掛けてやればよい。

ところで循環する無限小数のうち、ひときわ目を引くものがあるだろう。$0.999\cdots$ のことだ。これも同様に $x = 0.999\cdots$ とおいて

$$\begin{array}{rrcl} & 10x & = & 9.999\cdots \\ -) & x & = & 0.999\cdots \\ \hline & 9x & = & 9 \end{array}$$

としてみよう。あれ？ $x = \dfrac{9}{9}$ になったぞ。ということは $0.999\cdots = 1$ なんだろうか。その説明の前にぜひ 0.4999··· や 0.6999··· を分数にしてほしい。

雰囲気がつかめただろうか。話をちょっと前に戻すけれど、小数の濃度を調べているときに、すべての小数を $0.\alpha_1\alpha_2\alpha_3\alpha_4\cdots$ の形に表したね。このとき、0.5 のような小数は 0.5000··· とでもするのかなと考えなかっただろうか？ 実を言うと、そこには 0.5 や 0.5000··· のような、いわゆる有限小数は含まれていなかったのである。そこでは 0.5 は 0.4999··· の形で登場していたのだ。有限小数はすべて 999··· を含む、循環する無限小数になっていたのである。有限小数は無限小数で表記しておくと都合がよいのだ。そうしておけば、同じ数を 2 回数えることはなくなるから。

さて、話題は 0.999··· へ戻る。$0.999\cdots = 1$ である。なぜなら $\dfrac{9}{9}$ を実際に割り算してみると、0.999··· であることが確認できるのだから。

$$\begin{array}{rr|llll} & & 0. & 9 & 9 & 9 \\ \cline{3-6} 9 &) & 9. & 0 & 0 & 0 \\ & & 8 & 1 & & \\ \cline{3-5} & & & 9 & 0 & \\ & & & 8 & 1 & \\ \cline{4-6} & & & & 9 & 0 \end{array}$$

これは何だか不思議な計算だ。でも、あっている。このようなことが起こるのは、割り切れる割り算に 2 通りの表記法があるからだ。ひとつは素直に割り切ってしまう計算である。そうすれば $\dfrac{9}{9} = 1$ となる。そしてもうひとつは、今の例のように 999··· と商を立て続けてしまう計算である。こちらが濃度の話に登場した表記なのだ。

それにしても 999··· には悩まされそうだ。その底流をなすのは無限であることは間違いない。深みにはまる前に、循環小数の話へ戻ろう。

循環小数を計算してみると、$\dfrac{1}{6} = 0.1666666\cdots$ のように**循環節**が短いものと $\dfrac{1}{7} = 0.1428571\cdots$ のように循環節が長いものがある。$\dfrac{1}{6}$ は余りが最大で 5 種類出る可能性がある。$\dfrac{1}{6}$ は割り切れないので、0 は余りの種類に含めていない。このことは循環節が最大で 5 になる可能性があるわけだが、実際の循環節は 1 だ。ところが $\dfrac{1}{7}$ は循環節が最大で 6 になる可能性を持ち、その通り 6 の循環節を持っている。どんな有理数が、可能な限度を目一杯使うのだろう。いくら **VBA** が浮動小数点数を扱えるといっても、無限に小数点以下を計算してくれるわけではない。そんなときはどうしよう？

それでは、ある有理数がどれぐらいの長さの循環節を持つか調べよう。この段階のマクロとしては少々手強いかもしれない。でも、がんばって。

いちばんの問題は、循環節の長さの調べ方だ。コンピュータが無限に小数を表示してくれれば楽だが、そうはいかない。そこで発想を変えよう。小数は以前と同じ余りが出たときに循環を繰り返す。すると商を調べるのではなく、余りを調べればよいことに気付くだろう。しかも余りは整数値だから、変数は `Integer` 型を用意すれば十分だ。また、分数は分子が分母より小さいと決めつけてよい。なぜなら $\dfrac{22}{7}$ のような分数は、必ず $3+\dfrac{1}{7}$ のような形にできる。この場合、循環する鍵を握っているのは $\dfrac{1}{7}$ のような、分子が分母より小さい分数である。従って、分子が分母より大きい分数を考える必要はない。

programming list [RecurDec.vba]

```
1 : Sub RecurDec()
2 : Dim a, b, r, i As Integer
3 :
4 :     a = Sheet1.Cells(1, 1).Value
5 :     b = Sheet1.Cells(1, 2).Value
6 :
7 :     r = a
8 :     For i = 1 To (b - 1)
9 :         r = (r * 10) Mod b
10 :         Sheet1.Cells(1 + i, 1) = r
11 :     Next i
12 : End Sub
```

このマクロはいくつか重要なポイントがある。順に見ていこう。

まず、これはワークシートの A1, B1 セルに調べたい数の “分子”, “分母” が入力されていることが前提である。プログラムは 4: 行目と 5: 行目でワークシートの “分子”, “分母” を整数変数である `a`, `b` に代入している。さて、入力を受け入れた **VBA** は、割られる数 `a` と割る数 `b` の値を知ったことになる。今は割られる数が割る数より小さいので、割られる数はすでに余りになっていると考えてもよい。そこで `a` を `r` に代入し、分子を最初の余りとして扱っている。これが 7: 行目だ。

そして、いよいよ 8: 〜 11: 行目の `For-Next` 構文だ。繰り返しの条件が、`i = 1 To (b - 1)` となっていることに注意してもらいたい。`i` は `1` から始めて `b` の 1 つ前の数で終わる。`b` が分数の分母であったことを思い出してほしい。循環小数は分母の数より多い循環節を持たないので、分母 `b` の回数の割り算をするうちに必ず同じ余りが出るものだ。もし、そこまでに同じ余りが出現しなければ、その分数の循環節は許される最大の $b-1$ であることが分かるのである。

さらに、肝心な余りを求める計算は 9: 行目の `r = (r * 10) Mod b` で求めている。等号は両辺が等しいことを意味しない。前に述べたように、右辺の結果を左辺へ代入するのである。まず右辺だが、`(r * 10)` は余りを 10 倍している。これはわれわれが割り算をする際、上から 0 を下ろ

す操作である。続く `Mod b` は余りを求める演算である。コンピュータプログラムには色々な演算記号があるものだが、`Mod` はそのひとつだ。`A Mod B` と書いた場合、この演算は "`A` を `B` で割ったときの余り" を求める計算なのである。よって一連の `(r * 10) Mod b` は、余りの 10 倍をもう一度割って、次の余りを求める操作になっているわけだ。そして、次の余りを `r` に代入してやれば、再び同じ式でさらに次の余りを求められることになる。

10: 行目の `Sheet1.Cells(1 + i, 1) = r` 文は、もう説明の必要はないだろう。マクロはセルに余りを表示することになるが、`For` 文を繰り返す度に余りを入れるセル位置が下に進む。結果、A2 セル以下に余りの数が並ぶのである。$\dfrac{1}{113}$ の余りは興味深いと思う。調べるとよいだろう。

というわけで、与えられた分数の余りが次々と表示され、分数の循環節が長いか短いかは目にできるようになった。しかし、余りを眺めていてもさほど楽しいわけではない。やはり商を眺めて、その分数の特徴を知りたいだろう。だが旅は始まったばかりだ。われわれには、まだ知るべきことが山ほどある。ここの景色をじっくりと眺めるのは、帰り道で再びこの地に来たときにしよう。