0.4 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。

たとえば $\dfrac{1}{6}$ 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。

$$
\begin{array}{rrrrrr}
 & & 0. & 1 & 6 & 6 \\
\cline{3-6}
6 &) & 1. & 0 & 0 & 0 \\
 & & & 6 & & \\
\cline{3-5}
 & & & 4 & 0 & \\
 & & & 3 & 6 & \\
\cline{4-6}
 & & & & 4 & 0
\end{array}
$$

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出るからなのだ。

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

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

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 $0.4999\cdots$ や $0.6999\cdots$ 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 $0.5000\cdots$ 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 $0.5000\cdots$ 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 $0.4999\cdots$ の形で登場していたのだ。有限小数はすべて $999\cdots$ 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 $0.999\cdots$ 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、$0.999\cdots$ であることが確認できるのだから。

$$\begin{array}{r|rrrr} & 0. & 9 & 9 & 9 \\ \hline 9 & 9. & 0 & 0 & 0 \\ & 8 & 1 & & \\ \cline{2-3} & & 9 & 0 & \\ & & 8 & 1 & \\ \cline{3-4} & & & 9 & 0 \end{array}$$

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、いまの例のように $999\cdots$ 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 $999\cdots$ 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。道に迷う前に、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 $\frac{1}{7}$ 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。そんなときはどうしよう？

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出力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。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 $\frac{22}{7}$ のような分数は、必ず $3+\frac{1}{7}$ 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 $\frac{1}{7}$ のような、分子が分母より小さい分数である。したがって、分子が分母より大きい分数を考える必要はない。たとえば、次のマクロは $\frac{1}{7}$ の筆算余りを調べるものだ。

```
\newcommand\recurdec{%
  \newcount\r \newcount\b \newcount\c \newcount\q
  \r=1 \b=7 \c=1
  \loop
    \number\r\
    \multiply\r10
    \q=\r \divide\q\b
    \multiply\q\b
    \advance\r-\q
    \advance\c1
  \ifnum\c<\b \repeat
}
```

これで、『$\frac{1}{7}$ の筆算余りは\recurdec です。』と書けば『$\frac{1}{7}$ の筆算余りは 1 3 2 6 4 5 です。』が出力される。

まず分数は $\frac{a}{b}$ という形だが、いま考えている分数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いる。そこで $\frac{r}{b}=\frac{1}{7}$ と見て代入をしている。マクロで使う変数は、「$a \div b = q$, 余り r」の表現に合わせた。また、\c は繰り返し回数をカウントするための変数である。繰り返しの回数は 1 から数えるので\c=1 であるが、商\q は計算途中に代入されるので初期値は与えなかった。本当は\q=0 としておくのがよい。また、繰り返しの回数は最大でも分母の値を下回ることに注意しよう。そのため、割り算を繰り返す回数は b より小さくなる。

使ったのはお馴染みの\loop〜\repeat 命令である。他に掛け算と割り算をする必要があるので、\multiply と\divide の命令を使っている。使い方は\advance x a と同じで、\multiply x a なら x に a を掛けた値が新たに x に格納され、\divide x a なら x を a で割った値が新たに x に格納される。ただし、これらは整数値の計算が前提である。とくに割り算では、x を a で割った値とは整数値を指す。つまり、\divide\q\b は端数を捨てた整数値が\q に格納される。

では、マクロはどのように動作しているのだろう。実際に $1 \div 7$ を計算する場合を思い浮かべてみよう。

$$
\begin{array}{rr|llll}
 & & 0. & 1 & & \\
\hline
7 &) & 1. & 0 & 0 & 0 \\
 & & & 7 & & \\
\cline{3-4}
 & & & 3 & &
\end{array}
$$

この場合は、まず割られる数である 1 を 10 倍しておいて、そこから商にあたる 1 に割る数 7 を掛けたものを引いて、余りである 3 を求めている。この手順を、代入する変数を切り替えながら行ったのが上のマクロである。

まず、`\number\r\`で最初の余りである 1 が出力される。末尾の “\” は、次の数との間に空白を入れたいので記述した。数字をくっつけて出力するなら不要である。続けて`\multiply\r10` で余りである`\r` を 10 倍する。この値は、7 で割られる数であると同時に、ここから商に 7 を掛けた値を引く数でもある。1 数で 2 役をこなさなければならないので、一旦`\q=\r` として変数`\q` へ退避（コピー）しておく。そうすると心置きなく`\divide\q\b` で商を求めることができる。`\divide` による割り算は商が整数値になる。つまり $14 \div 5$ であれば、それは 2.8 になるのではなく 2 になる。だから、ここで`\q` に入っている値は純粋に整数値の商である。ちなみに、TEX には商を小数値で求める命令はない。

商が分かれば 7 を掛けて引けばよい。それが一連の`\multiply\q\b`、`\advance\r-\q` である。TEX は引き算を負の値の足し算で行うので、`\advance\r-\q` は $r - q$ と同じことである。これで`\r` に新たな余りが入ったので、以下はこの繰り返しとなる。

繰り返しの範囲が $b - 1$ までででよいことに注意してもらいたい。`\b` は分数の分母であった。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` の回数だけ割り算をすれば必ず同じ余りが出る。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。

ところで、余り 3 にたどり着くまでに行った計算は

$$1 \times 10 - 1 \times 7 = 3$$

である。筆算においても、一般的なプログラミング言語においても、このような計算は 1 行で済むことが多い。しかし、TEX のマクロではそうはいかないのである。なぜなら、一般的なプログラミング言語では**演算子**—すなわち $+$, $-$, $\times$, $\div$ などの計算記号—を混在させて使えるのに対し、

TEX には演算子がない。あるのは命令である。命令は混在させて使うものではなく、1 回ずつ使うものである。そのために、以上のような回りくどい計算をするマクロになったわけなのだ。

　というわけで、与えられた分数の余りが次々と出力され、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。だがいまは道を這い回り始めたばかりだ。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。