

## 7.4 円周率 112 桁

$\pi$  の近似計算の開拓は任せたけれど、道筋はつけておこう。その道を舗装するのは君たちの仕事だ。公的な事業費は出せないが。112 桁というのは何とも中途半端であるが、使えるレジスタに限りはあるし、それに 112 桁でも十分に変な。まず、変数と配列をレジスタに割り当てよう。

```
%変数
\newcount\n \newcount\r \newcount\cntr \newcount\upf \newcount\down
\newcount\arraysize \newcount\arraysizePlus
%配列
\newcount\ARY
\newcount\I \newcount\II \newcount\III \newcount\IV
\newcount\V \newcount\VI \newcount\VII \newcount\VIII
\newcount\IX \newcount\X \newcount\XI \newcount\XII
\newcount\XIII \newcount\XIV \newcount\XV \newcount\XVI
\newcount\XVII \newcount\XVIII \newcount\XIX \newcount\XX
\newcount\XXI \newcount\XXII \newcount\XXIII \newcount\XXIV
\newcount\XXV \newcount\XXVI \newcount\XXVII \newcount\XXVIII
\newcount\TOP \newcount\zero
\newcount\i \newcount\ii \newcount\iii \newcount\iv
\newcount\v \newcount\vi \newcount\vii \newcount\viii
\newcount\ix \newcount\x \newcount\xi \newcount\xii
\newcount\xiii \newcount\xiv \newcount\xv \newcount\xvi
\newcount\xvii \newcount\xviii \newcount\xix \newcount\xx
\newcount\xxi \newcount\xxii \newcount\xxiii \newcount\xxiv
\newcount\xxv \newcount\xxvi \newcount\xxvii \newcount\xxviii
```

うへっ！ 変数だけでこんなに使うの？ だけど仕方なかったんだ。他により方法が思いつかなかったからね。変数と配列がこれだけあると\ifcaseでの分岐も大量になる。いくつか必要になったのでパッケージにしてある。\\ifcaseで分岐するところでは以下のものが使われる。

```
%\ifcase 分岐
\newcommand\caseARYI{\ifcase\ARY%case0 is null
\or\I \or\II \or\III \or\IV \or\V
\or\VI \or\VII \or\VIII \or\IX \or\X
\or\xi \or\xii \or\xiii \or\xiv \or\xv
\or\xvi \or\xvii \or\xviii \or\xix \or\xx
\or\xxi \or\xxii \or\xxiii \or\xxiv \or\xxv
\or\xxvi \or\xxvii \or\xxviii
\fi}

\newcommand\caseARYII{\ifcase\ARY%case0 is null
\or\II \or\III \or\IV \or\V \or\VI
\or\VII \or\VIII \or\IX \or\X \or\xi
\or\xii \or\xiii \or\xiv \or\xv \or\xvi
\or\xvii \or\xviii \or\xix \or\xx \or\xxi
\or\xxii \or\xxiii \or\xxiv \or\xxv \or\xxvi
\or\xxvii \or\xxviii \or\TOP
\fi}
```

```

\newcommand\caseARYi{\ifcase\ARY%case0 is null
\or\i      \or\ii     \or\iii    \or\iv     \or\v
\or\vi     \or\vii    \or\viii   \or\ix     \or\x
\or\xi     \or\xii    \or\xiii  \or\xiv    \or\xv
\or\xvi    \or\xvii   \or\xviii \or\xix    \or\xx
\or\xxi    \or\xxii   \or\xxiii \or\xxiv   \or\xxv
\or\xxvi   \or\xxvii  \or\xxviii
\fi}

\newcommand\caseARYzero{\ifcase\ARY%case0 is null
\or\zero   \or\i      \or\ii     \or\iii    \or\iv
\or\v      \or\vi     \or\vii    \or\viii   \or\ix
\or\x      \or\xi     \or\xii    \or\xiii   \or\xiv
\or\xv     \or\xvi    \or\xvii   \or\xviii  \or\xix
\or\xx     \or\xxi    \or\xxii   \or\xxiii  \or\xxiv
\or\xxv    \or\xxvi   \or\xxvii
\fi}

```

ところで配列は使い回すので初期化が欠かせない。使い回すと言っても  $\frac{1}{5^{2n-1}}$  の計算で使用したあと  $\frac{1}{239^{2n-1}}$  の計算に切り替えるときだけである。しかし直前の  $\frac{1}{5^{2n-1}}$  の計算で配列の値は実は全部 0 になってしまうので配列の初期化は実質的に不要だ。

続いて加算、減算、除算のルーティンである。

```

%加減貸借ルーティン
\newcommand\giveandtake{
\ifnum\caseARYI>9999
\upf=\caseARYI \divide\upf by10000
\global\advance\caseARYII by\upf
\multiply\upf by10000
\global\advance\caseARYI by-\upf
\fi}

%加算
\newcommand\advanceIi{\ARY=1
\loop \ifnum\ARY<\arraysizePlus
\global\advance\caseARYI by\caseARYi
\giveandtake
\advance\ARY by1 \repeat
}

%減算
\newcommand\subtractIi{\ARY=\arraysize
\loop \ifnum\ARY>0
\global\advance\caseARYII by-1
\global\advance\caseARYI by10000
\global\advance\caseARYI by-\caseARYi
\giveandtake
\advance\ARY by-1 \repeat
}

```

```
%除算
\newcommand\divideby[1]{\ARY=\arraysize
\loop \ifnum\ARY>0
\ r=\caseARYi
\ dwn=\r \divide\r by#1
\caseARYi=\r
\multiply\r by#1 \advance\dwn by-\r \multiply\dwn by10000
\advance\caseARYzero by\dwn
\advance\ARY by-1 \repeat
}
```

加減貸借ルーティンというのは、繰り上がりや繰り下がりするとき配列どうして貸し借りするための処理である。どれも簡単なルーティンに見えるが、すべて\ifcaseの分岐パッケージを用いて記述を簡略化している。

それと表示用のルーティンも必要だ。出力は正しく表示されなくてはならない。

```
%表示
\newcommand\fchk[1]{%
\ifnum#1<10 000\else\ifnum#1<100 00\else\ifnum#1<1000 0\fi\fi\fi}
\newcommand\dispI{%
\ifnum\TOP>0 \number\TOP.\fi
\ARY=\arraysize
\loop \ifnum\ARY>0
\ fchk\caseARYI
\ number\caseARYI\
\advance\ARY by-1 \repeat
}
```

配列に格納された数をそのまま出力すると、4桁の数が格納されていれば問題ないが、そうでないときは0が表示されなくなってしまう。fchkの書き方は微妙だろうが、これでうまく0が補える。さあ、ようやく準備が整った。πの計算をしよう。πの計算に使う式はだいたい以前に目にしたマチンの式である。あらかじめ4倍した

$$\pi = 16 \left( \frac{1}{1 \cdot 5} - \frac{1}{3 \cdot 5^3} + \frac{1}{5 \cdot 5^5} - \cdots \right) - 4 \left( \frac{1}{1 \cdot 239} - \frac{1}{3 \cdot 239^3} + \frac{1}{5 \cdot 239^5} - \cdots \right)$$

を使うことにする。そしてマクロ計算の都合上、初項から常に $5^2$ や $239^2$ で割り続けたいので、

$$\pi = \frac{80}{5} \left( \frac{1}{1 \cdot 5} - \frac{1}{3 \cdot 5^3} + \frac{1}{5 \cdot 5^5} - \cdots \right) - \frac{956}{239} \left( \frac{1}{1 \cdot 239} - \frac{1}{3 \cdot 239^3} + \frac{1}{5 \cdot 239^5} - \cdots \right)$$

としておいた。xxviii=800000とxxviii=9560000は、このことから出た数値だ。これで初期値から常に $5^2$ や $239^2$ で割ることが可能になる。当初の式とは意味合いが少々変わっても、やはりマクロが書きやすい方がよい。もっとも、このために後日マクロを見直したときに意味不明のルーティンに出会うこともしばしばある。必要なコメントは本来コードと一緒に記述しておくべきである。

$\pi$  の計算をしたあとは表示ルーティン `\dispI` が面倒を見てくれる。

```
%pi の計算
\makeatletter
\arraysize=28 \arraysizePlus=29
\xxviii=800000 \cntr=1 \n=1
\@whilenum\cntr<80 \do{%(1/n)(5^{2n-1})<10^{-112}より
  \divideiby{25}
  {\divideiby\n
  \ifodd\cntr \advanceIi \else \subtractIi \fi}
\advance\cntr by1 \advance\n by2}
%
\xxviii=9560000 \cntr=1 \n=1
\@whilenum\cntr<24 \do{%(1/n)(239^{2n-1})<10^{-112}より
  \divideiby{57121}
  {\divideiby\n
  \ifodd\cntr \subtractIi \else \advanceIi \fi}
\advance\cntr by1 \advance\n by2}
\makeatother
```

これで、『円周率を 112 桁出力すると `\dispI` である。』と書けば『円周率を 112 桁出力すると  
3.1415 9265 3589 7932 3846 2643 3832 7950 2884 1971 6939 9375 1058 2097 4944 5923 0781 6406  
2862 0899 8628 0348 2534 2117 0679 8214 8086 5127 である。』が出力される。

はあ？ たったこれだけ？

あんなに苦勞してマクロを書いても出力はほんの 2 行ですか。しかも最後の 2 桁は誤差を含んでいるし。これならどこかのサイトからコピペすれば十分だ。脱力感とともに蝶道から退散するでしょう。