

## Jで数式を美しく表わしてみよう 自然数の和

西川 利男

コロナ禍のため、毎日家にいることをよぎなくされている。ごく最近つぎのような数学の本を入手した。[1]

[1] 横山明日希「数式図鑑－美しく、役に立つ科学の宝石箱」  
ブルーバックス B-2178 講談社(2022)

Jで難しい計算を行うのはもちろんだが、計算途中や結果をきれいに表示し、印刷することにもJは極めて有効である。

数式図鑑の第1ページに下のようなきれいな計算が載っている。  
つれづれなるままに、これをJでプログラムをやってみた。こんなもの、たいしたことないとかをくくっていたが、なかなかこずった。

かについては稿

和が美しく図形のように連なる感動

自然数の和

$$1+2=3$$

$$4+5+6=7+8$$

$$9+10+11+12=13+14+15$$

なぜ、こうなる  
末に説明がある。

## 1. Jによる自然数の和

まず個別に1行ずつやった結果は、次のようになる。

func 1  
 $1+2=3$

func 4  
 $4+5+6=7+8$

func 9  
 $9+10+11+12=13+14+15$

func 16  
 $16+17+18+19+20=21+22+23+24$

func 25  
 $25+26+27+28+29+30=31+32+33+34+35$

func 36  
 $36+37+38+39+40+41+42=43+44+45+46+47+48$

func 49  
 $49+50+51+52+53+54+55+56=57+58+59+60+61+62+63$

func 64  
 $64+65+66+67+68+69+70+71+72=73+74+75+76+77+78+79+80$

func 81  
 $81+82+83+84+85+86+87+88+89+90=91+92+93+94+95+96+97+98+99$

これをまとめて、表示すればよい。

このJプログラムは後に詳細に述べる。

## 2. 問題点と個別の計算表示

上では1つの関数で行ったが、ここにいたるまでは、いろいろな試行錯誤でプログラムを作った。

計算はたいしたことはないが、桁数が2桁だけならよいが、1桁のもの、1桁と2桁との混じったものといろいろある。しかたがないので、まず、別べつに処理することにした。

以下のJプログラムfは先頭の数すべて16、25…など2桁である場合のプログラムとその実行結果である。

なお、実行の途中経過を示すため、細かく出力している。

```
wr =: 1!:2&2

NB. f 16
NB. f 25
NB. f 36
f =: 3 : 0
NB. 'a n' =. y.
a =. y.
n =. (%: a) +1
wr a, n,
AA =. ''
i =. 0
while. i < n
do.
  wr i, (a + i)
  wr A =. a * i + (i + 1)
  wr AA =. AA , " : a + i
  i =. i + 1
end.
AAA =: AA
AAAA =. } : , ((-: #AAA), 2)$AAA , "(1)'+
wr AAAA
wr '====='
b =. a + i
BB =. ''
j =. 0
while. j < (n-1)
do.
  wr j, (b + j)
  wr B =. b * j + (j + 1)
  wr BB =. BB , " : b + j
  j =. j + 1
end.
BBB =: BB
BBBB =. } : , ((-: #BBB), 2)$BBB , "(1)'+
wr BBBB
wr AAAA, '=', BBBB
'*** end ***'
)
```

```

    f 16
16 5
0 16
16
16
1 17
48
1617
2 18
80
161718
3 19
112
16171819
4 20
144
1617181920
16+17+18+19+20
=====
0 21
21
21
1 22
63
2122
2 23
105
212223
3 24
147
21222324
21+22+23+24
16+17+18+19+20=21+22+23+24
*** end ***

```

ここで、J独特の演算をしると次のようなものである。

```

%: 4 ⇒ 2          平方根を得る
": 3.14 ⇒ 3 . 1 4    数値を4つの文字列として得る
-: 6 ⇒ 3 ⇒ 3        1/2にする
AAA =: 2, 3, 4, 5, 6, 7   とするとき
# AAA ⇒ 6            要素の数を返す
((-: # AAA), 2) $ AAA    数値AAAを3行2列の配列に変形する
2 3
4 5
6 7

```

などである。

このように1桁の場合、1桁2桁の混じった場合はそれぞれ、別べつのプログラムとして作らざるを得なかった。

1つのプログラムとして、処理するには細かい点は省略するが、出力のフォーマットをつぎのように変えることが必要である。

1桁の場合

```
AAAA1 =. } : , ( . AAA ) , "(1) '+'
```

```
BBBB1 =. } : , ( . BBB ) , "(1) '+'
```

1桁、2桁が交じった場合

```
AAAA2 =. } : , ((4, 2)$ ' ', AAA) , "(1)'+
```

```
BBBB2 =. } : , (((-: #BBB), 2)$BBB) , "(1) '+'
```

これらを考慮して引数として最初の数、1、4、9、16、… を入れると、先のページ2のような結果を出力する最終の関数 func は次のようになる。

```
func =: 3 : 0
NB. 'a n' =. y.
a =. y.
n =. (%: a) + 1
AA =. ''
i =. 0
while. i < n
  do.
    i, (a + i)
    A =. a * i + (i + 1)
    AA =. AA , ": a + i
    i =. i + 1
  end.
AAA =: AA
if. a >: 16
  do.
    AAAA =. } : , (((-: #AAA), 2)$AAA) , "(1)'+
  else.
    if. a = 9 do. AAAA =. } : , ((4, 2)$ ' ', AAA) , "(1)'+ end.
    if. a = 4 do. AAAA =. } : , ( . AAA ) , "(1)'+ end.
    if. a = 1 do. AAAA =. } : , ( . AAA ) , "(1)'+ end.
  end.
b =. a + i
BB =. ''
j =. 0
while. j < (n-1)
  do.
    j, (b + j)
    B =. b * j + (j + 1)
    BB =. BB , ": b + j
    j =. j + 1
  end.
BBB =: BB
if. a >: 16
  do.
    BBBB =. } : , (((-: #BBB), 2)$BBB) , "(1)'+
  else.
    if. a = 9 do. BBBB =. } : , ((3, 2)$BBB) , "(1)'+ end.
    if. a = 4 do. BBBB =. } : , ( . BBB ) , "(1)'+ end.
    if. a = 1 do. BBBB =. } : , ( . BBB ) , "(1)'+ end.
  end.
RES =. AAAA, '= ', BBBB
'*** end ***'
)
```

```
run =: 3 : 0
```

```
N =. y.
```

```
i =. 1
```

```
while. i <: N
```

```
do.
```

```
ii =. *: i
```

```
res =: func ii
keta =. res i. '='
wr ((30-keta)#' '), res
i =. i + 1
end.
'*** end ***'
)
```

#### 4. 最終結果

さらに最終結果として出力するプログラム run は、各行から文字イコール(=)の位置をカウントして、それを中央にもっていくため、各行の先頭に文字スペースを付加したものである。

イコール(=)を中心にきれいな整数計算の図形として出力した結果はつぎようになる。

```
run 9

      1+2=3
     4+5+6=7+8
    9+10+11+12=13+14+15
   16+17+18+19+20=21+22+23+24
  25+26+27+28+29+30=31+32+33+34+35
 36+37+38+39+40+41+42=43+44+45+46+47+48
49+50+51+52+53+54+55+56=57+58+59+60+61+62+63
64+65+66+67+68+69+70+71+72=73+74+75+76+77+78+79+80
81+82+83+84+85+86+87+88+89+90=91+92+93+94+95+96+97+98+99
*** end ***
```

#### 5. おわりに

計算処理としては、たいしたことをやったわけでない。しかし、1桁、2桁、1桁2桁が混じったもの、それらを1つにまとめて行うプログラムは試行錯誤でもなかなか大変だった。

ボケ予防の脳トレのJプログラムとでもいうべきか。数日間楽しみ、苦しみ、やっと出来上がったというのが本音である。

## 数式図鑑からの説明

自然数を並べていくだけで、美しい式が作れることをご存じでしょうか？ 2つの連続する自然数を足すと次の自然数に、3つの連続する自然数を足すと、次に連続する2つの自然数の和に……というように、左辺の項数と右辺の項数の差が1になった自然数の式が連続して現れます。どうしてこのような式が成立するのかを考えてみましょう。

よく見ると、いちばん左の項は1, 4, 9と、平方数(整数の2乗となっている数)になっていることがわかります。

2行目と3行目の式を例に考えると、 $4=2+2$ ,  
 $9=3+3+3$ となるので、

$$2+2+5+6=7+8$$

$$3+3+3+10+11+12=13+14+15$$

と書き換えることができます。ここで上の式では2つの2を5, 6にそれぞれ足すと7, 8になり、下の式では3つの3を10, 11, 12にそれぞれ足すと13, 14, 15になります。つまり、右辺と同じ式になることがわかりますね。

では、もう少し一般化して考えてみましょう。

冒頭の式は、いちばん左の項を $a \times a$ の平方数として、

$$(a \times a) + (a \times a + 1) + (a \times a + 2) + \cdots + (a \times a + a) \\ = (a \times a + a + 1) + (a \times a + a + 2) + \cdots + (a \times a + a + a)$$

と表すことができます。左辺の項数は全部で $(a+1)$ 個、右辺の項数は全部で $a$ 個です。ここで、いちばん左の項を $(a \times a) = (a + a + a + \cdots + a)$ と変形して、各項にそれぞれ $a$ を足していくと、左辺は、

$$(a \times a + a + 1) + (a \times a + a + 2) + \cdots + (a \times a + a + a)$$

となります。この式は、右辺とまったく同じ式になりますね。このような式変形を試みることで、隠された美しい法則性が見えてくるのが数式にはよくあります。