

こだわりのプログラミング

真の値を求めて

北海道苫小牧東高等学校

矢嶋 裕之

はじめに

前回の数学実践研究会では、数多くのレポートが集まり、最後の方は超特急でした。私の意味のないレポートは構わないものの、最後の中村先生のレポートはゆっくり説明が聞けず残念でした。その前回の数実研から帰って数日後、数学の好きな生徒と教室で教科書には載っていないような数学の題材について話をしていた時のことです。常用対数を利用して 2^{50} が何桁かというところを指して、『先生、ところでこの 2^{50} って、どんな数なのですか。』と聞いてきました。確かに 2^{50} という数は桁数の話としては使いますが、(真の値が大きすぎることもあって) 学校では桁数だけの説明したら次の内容に入ってしまう。生徒は漠然とそこが疑問になったようです。そういう場合、『とても大きい数だよ』では答えになりませんよね。『だったら、一緒に求めてみよう』という方向になりました。そう、これが『こだわりのプログラミング』と私が大学時代に言っていた実用計算の世界の話なのです。本稿は、一見簡単に思われがちな計算でもオーバーフローという状況の中で、真の値が求めにくいプログラミングについて、実用計算となるものについて『 $n!$ 』と『 a^n 』の計算を例にまとめたものです。

第1章 単純計算ではいけない

$n! = n \times (n - 1) \times (n - 2) \times \dots \times 3 \times 2 \times 1$ のことであり、 $a^n = a \times a \times a \times \dots \times a \times a$ (a を n 回掛ける) の計算であることは自明です。しかし、このままの形で(リスト1のように) BASIC プログラムを組むと一見問題ないように見えますが、すぐにオーバーフローしてしまいます。実際、このリスト1の状況では n の値が 17 までは階乗の値(真の値)を表示しますが、18 からは指数表示になってしまいました(十進 BASIC 利用時)。

【 リスト1 : $n!$ の値 】

```
10 input n
15 kotae=1
20 for i=n to 1 step -1
30 kotae=kotae*i
40 next i
50 print n;" ! の値 = ";kotae
60 END
```

どんな値（真の値）なのかを知りたいという要望に対してそれをかなえるプログラム、それが今回のレポートにつながったのです。

第2章 実際に使えるプログラム

前章で述べたように流れ図もプログラムも正しくても本当の値（概数ではなく）を表示してくれないプログラムでは今回の生徒のような要望には応えられません。大学時代の私のノートにはこうした実用計算に関する記述がたくさんあります。そこで、前述の通り、 $n!$ と a^n について実用計算となるものの一例をここでふれたいと思います。

【 リスト2： $n!$ の実用計算 】

```
100 defint a-z
110 dim mx(100)
120 dim lx(100),kx(100)
130 dim moji(100,4)
131 for i=1 to 100
132 for j=1 to 4
133 moji(i,j)=0
134 next j
135 next i
140 input nn
150 for n=2 to nn
160 print
180 print n;"! の値は次の通りである。"
190 print
200 no=1
210 kx(1)=1
220 for i=2 to n
230 noo=no+1
240 if noo>100 then goto 890
250 for l=1 to noo
260 mx(l)=0
270 lx(l)=0
280 next l
290 kx(noo)=0
300 nl=0
310 nl=nl+1
320 ind=0
330 if nl>no then 420
340 for j=1 to i
350 lx(nl)=lx(nl)+kx(nl)
```

```
360 if lx(nl)>9999 then gosub 900
370 next j
380 lx(nl)=lx(nl)+mx(nl)
390 if lx(nl)>9999 then gosub 900
400 no=no+ind
410 go to 310
420 for j=1 to no
430 kx(j)=lx(j)
440 next j
450 next i
451 ccc=no/2
460 jjj=int(ccc)
470 if jjj=0 then go to 540
480 for ji=1 to jjj
490 kk=kx(ji)
500 jj=no-ji+1
510 kx(ji)=kx(jj)
520 kx(jj)=kk
530 next ji
540 lj=4
542 zzz=kx(1)/10
550 mm=int(zzz)
570 if mm=0 then goto 620
580 moji(1,lj)=kx(1)-mm*10
590 kx(1)=mm
600 lj=lj-1
610 go to 542
620 moji(1,lj)=kx(1)
630 if no=1 then goto 800
640 for ll=2 to no
650 llj=4
660 sss=kx(ll)/10
665 mm=int(sss)
680 if mm=0 then goto 730
690 moji(ll,llj)=kx(ll)-mm*10
700 kx(ll)=mm
710 llj=llj-1
720 goto 660
730 moji(ll,llj)=kx(ll)
740 if llj=1 then goto 790
```

```

750 llj=llj-1
760 for mll=1 to llj
770 moji(ll,mll)=0
780 next mll
790 next ll
800 for ik=lj to 4
810 print moji(1,ik);
820 next ik
830 for iik=2 to no
840 for iij=1 to 4
850 print moji(iik,iij);
860 next iij
870 next iik
871 print
880 next n
890 END
900 zxzx=lx(nl)/10000
905 is=int(zxzx)
910 ia=lx(nl)-is*10000
920 if no=nl then ind=1
921 vv=nl+1
930 mx(vv)=mx(vv)+is
940 lx(nl)=ia
950 return

```

【 リスト 3 : a^n の値の実用計算、10000 桁まで可能型 】

```

100 dim k(10000)
110 input a
120 input n
121 print a;"の";n;"乗の値は次の通りである"
122 print
130 LET i=1
131 z=0
132 LET k(i)=a
150 if n=1 then goto 300
160 if k(i)>9 then gosub 400
170 for l=2 to n
172 for i=10000 to 1 step -1
180 LET k(i)=k(i)*a
181 next i

```

```

183 for i=1000 to 1 step -1
190   if k(i)>9 then gosub 400
192 next i
200 next l
210 goto 600
300 print k(i)
310 goto 630
400 u=k(i)/10
410 iu=int(u)
420 k(i)=k(i)-iu*10
430 m=i+1
440 k(m)=k(m)+iu
441 if k(m)>9 THEN goto 700
450 return
600 for i=1000 to 1 step -1
602 if k(i)=0 and z=0 then goto 620
610 print k(i);
611 z=z+1
620 next i
630 END
700 u2=k(m)/10
710 iu2=int(u2)
720 k(m)=k(m)-iu2*10
730 m=m+1
740 k(m)=k(m)+iu2
750 if k(m)>9 THEN goto 700
760 return

```

このプログラム以外のものでも同様な計算をする実用プログラムは多数あると思います。要はこれらの計算をどのようにして行うかという点（アルゴリズム）の問題であり、様々な考え方があるのです。

第3章 十進BASICを使ってみて

今回のリスト1からリスト3までのプログラムは学校のパソコンで実行したので、BASIC言語としては『十進BASIC』を利用しました。実は、今までの『N88BASIC』と異なり、印字部分のコマンドで『;』で各数字をつなげていくとき、『十進BASIC』ではすこし字間が空くようです。そのため、桁数が多くなるにつれて（見た目的に）やや気になり始めました。マニュアルをみながら改善できないか今後考えてみたいと思います。また、前回のレポートでもふれましたが、『十進BASIC』ではオプションで文法の切り替えができません。今回のリスト1からリスト3のプログラムは『Microsoft

Basic 互換』では正常に実行しますが、『旧 JIS 基本 BASIC 互換』や『JIS FULL BASIC』ではエラーとなるものもありました。こうした点にも気をつけながら、数値計算を実行していかなければならないと思います。OS が Windows になり、『N 8 8 B A S I C』等昔の B A S I C 言語が使えなくなりやや不便ですが、『十進 B A S I C』など今のパソコンでも使えるものを利用して今後もやっていきたいと思っています。

おわりに

今回のレポートは、『はじめに』でもふれたように生徒の一言に端を発したものではありませんが、時岡先生（札幌拓北高等学校）が私が前任校時代に書いた『n! の話し』というタイトルのレポートについて問い合わせをしてくれなかったら、このようにレポートとしてまとめることはなかったと思います。大学時代に流れ図にもプログラミングにも問題がないのに実用には全然ならなかった（すぐオーバーフローを起こして）プログラムを改良して汎用性を持たせた『こだわりのプログラミング』術、最近再度やってみて奥の深いことも実感できました。これまでは、『数学の先生だから』とあまりそうしたプログラミングについて何かを考えたり教えたりするといったことはありませんでした。しかし、今後の『情報』の授業の実践をも考えてみたとき、時間のあるときに復習してみるのもいいかなと思いました。ただ、今回、このレポートで紹介したような『プログラミング』は実用的ではありますが、教室で生徒に教えるものとは異なります。そのバランス（『机上のプログラミング』か『実用のプログラミング』かの）をうまく保つことは大切なことだと思います。今後は、この研究会でも新教科『情報』にも通じるような話題や実践がでてきたら幸いですと願いつつペンを置きます。

2月18日夜

パソコン教室にて