

多倍長演算のプログラムについてのレポート

金沢大学 4年 数物化学類

荒井 雄太

・プログラム作成の経緯と結果

ベルヌーイ数を求めるプログラムを作成していた際、途中で出てくる数が大きすぎてプログラムが扱える数の範囲を超えていた為に正確な計算ができなかったので、大きな桁数の演算でも扱える方法を考えた結果、今回の多倍長演算のプログラム作成に至った。今回は多倍長演算を用いた四則演算のプログラムの作成を試み、結果、除法を除く四則演算のプログラムを作成できた。

・t 桁の自然数 a を初期値として入力するプログラム

```
integer n,t
parameter(n=30)
parameter(t=20)
integer a(n),a1(t),m
character*30 shoki
shoki='12345678901234567890'
do m=1,n
  a(m)=0
enddo
do m=1,t
  a1(m)=0
enddo

do m=1,t
  a1(t+1-m)=ichar(shoki(m:m))-48
  a(t+1-m)=a1(t+1-m)
enddo
```

n : 入力できる桁の最大値
t : 入力したい自然数 a の桁数
shoki : 半角 n 文字までの文字が入る型
(shoki の中に、入力したい数値を「文字列として」入力する)
ここでは
a=12345678901234567890
としている

最初は全ての配列に 0 を入力しておく
a(m) ($1 \leq m \leq n$) に a の m 桁目の数が入るようにしていく

ここからが数値入力の部分

ichar は型の特典部分の文字をコード化するコマンド。
ここでは、shoki の中の「左から数えて」 m 番目から m 番目までの文字(つまり左から m 番目の文字)を文字コード(文字や記号をコンピュータで認識するために割り当てられた固有の数字)に変換している。

半角数字の文字コードは、0には48が割り当てられていて、数字が1増える毎に文字コードも1増えるので、変換後の数から48を引けば元の数字に復号できる。
m 桁目の数が a(m)の箱に入る様に do 文を組む

```
do m=n,1,-1
  write(*,*) ,a(m)
enddo
stop
end
```

入力結果をリスト表示する

プログラム終了

実行結果

```
0
0
0
0
0
0
0
0
0
0
0
1
2
3
4
5
6
7
8
9
0
1
2
```

3
4
5
6
7
8
9
0

a=12345678901234567890
が a(n)~a(1)に格納できた

今後の計算では a,b 2つの数を初期値として入力する必要があり、また、a,b は整数なのでそれぞれの符号も入力する必要があるので、初期値入力と同時に符号も入力できるようにしたのが以下のプログラム

・ a,b の初期値入力用のプログラム(a,b ∈ Z)

```
integer n,s,t
parameter(n=20)
parameter(s=3)
parameter(t=2)
integer a(n),a1(s),m
integer b(n),b1(t),ahugou,bhugou,aa,bb
character*20 ashoki
character*20 bshoki
ahugou=-1
ashoki='276'
bhugou=1
bshoki='21'
do m=1,n
  a(m)=0
  b(m)=0
enddo
do m=1,s
  a1(m)=0
enddo
do m=1,t
  b1(m)=0
```

s : a の桁数
t : b の桁数
ahugou : a の符号を 1 か -1
で入力する
bhugou : b の符号を 1 か -1
で入力する
ここでは a = -276、b = 21
としている
(以後のプログラムでは
aa : a の符号、
bb : b の符号として扱う)

```
enddo
```

```
call nyuuryku(ashoki,bshoki,ahugou,bhugou,a,b,aa,bb,n,s,t)
```

入力のための部分を
サブルーチン化

```
write(*,*) , 'a hugou=',aa
```

```
do m=n,1,-1
```

a の入力結果を出力

```
  write(*,*) ,a(m)
```

```
enddo
```

```
write(*,*) , 'b hugou=',bb
```

```
do m=n,1,-1
```

b の入力結果を出力

```
  write(*,*) ,b(m)
```

```
enddo
```

```
end
```

プログラムのメイン部分終了

以下はサブルーチンのプログラム部分

```
subroutine nyuuryku(ashoki,bshoki,ahugou,bhugou,a,b,aa,bb,n,s,t)
```

```
integer n,s,t
```

```
integer a(n),a1(s),aa,ahugou
```

```
integer b(n),b1(t),bb,bhugou,m
```

```
character*20 ashoki
```

```
character*20 bshoki
```

```
aa=ahugou
```

a の符号 aa に±1 を入力

```
bb=bhugou
```

b の符号 bb に±1 を入力

```
do m=1,s
```

```
  a1(s+1-m)=ichar(ashoki(m:m))-48
```

```
  a(s+1-m)=a1(s+1-m)
```

```
enddo
```

```

do m=1,t
  b1(t+1-m)=ichar(bshoki(m:m))-48
  b(t+1-m)=b1(t+1-m)
enddo

end

```

サブルーチン部分終了

プログラム終了

以後は、四則演算を行う
 為のプログラムの説明を
 していく

- ・ 加法の多倍長演算のプログラム(a+b=c の c を求めるプログラム)

```

integer n,s,t
parameter(n=10)
parameter(s=3)
parameter(t=3)
integer a(n),b(n),c(n),e(n),aa,bb,cc
integer m,k,x
integer a1(s),b1(t),ahugou,bhugou
character*10 ashoki
character*10 bshoki

```

cc : c の符号部分

```

ahugou=1
ashoki='676'
bhugou=-1
bshoki='678'

```

ここでは a=676、b=-678
 として、a+b の和 c を求める

```

do m=1,n
  a(m)=0
  b(m)=0
  c(m)=0
enddo

```

```
do m=1,s
  a1(m)=0
enddo
```

```
do m=1,t
  b1(m)=0
enddo
```

c 初期値入力

```
call nyuuryku(ashoki,bshoki,ahugou,bhugou,a,b,aa,bb,n,s,t)
```

↑ここまでが初期値入力の部分
以降の行で加法の演算を
行っていく

*

```
if(aa.eq.bb) then
  cc=aa
  do m=1,n
    c(m)=a(m)+b(m)+c(m)
    if(c(m).ge.10) then
      c(m)=c(m)-10
      c(m+1)=c(m+1)+1
    endif
  enddo
else
```

a と b が同符号の時は
c の符号 cc=aa(=bb)となり、
a,b の絶対値同士を足せばよい
1~n 桁目毎に a(m)と b(m)
を足して c(m)に入力していき、
和が 10 を超えたら
c(m)から 10 を引いて c(m+1)
に 1 を加えて繰り上がりの処理
をする

```
else
```

a,b が異符号の時は、
①a,b の絶対値の比較
②大きい絶対値の符号を c の
符号にする
③大きい方の絶対値から
小さいほうの絶対値を引く
という順序で計算していく

```

10      k=n
        x=a(k)-b(k)
        if((k.eq.1).and.(x.eq.0)) then
            cc=1
            do m=1,n
                c(m)=0
            enddo

        else if((k.ne.1).and.(x.eq.0)) then
            k=k-1
            go to 10

        else

            cc=aa

            if (x.lt.0) then
                cc=bb
                do m=1,n
                    e(m)=a(m)
                enddo
                do m=1,n
                    a(m)=b(m)
                enddo
                do m=1,n
                    b(m)=e(m)
                enddo
            endif

```

以下は絶対値の比較

n 桁目から a(k)と b(k)の差を計算していき、

- n~1 桁目まで全ての桁で $a(k)-b(k)=0$ のときは $|a|=|b|$ なので、 $a+b=0$ が分かる

- $a(k)-b(k)=0$ (k 桁目で $|a|$ と $|b|$ の大小が分からないときは k-1 桁目を比較する

- $a(k)-b(k)\neq 0$ (a と b の大小が分かった時は、

まず $|a|>|b|$ であると仮定し、cc に aa を入力

$x<0$ (つまり $|a|<|b|$)のときは cc に bb を入れ直して、 $a(n)\sim a(1)$ の数値と $b(n)\sim b(1)$ を入れ替える

これで、 $|a|-|b|$ を「絶対値の大きい方」から「絶対値の小さい方」を引く計算として常に考えることができる


```

do m=1,n
  c(m)=a(m)-b(m)+c(m)
  if (c(m).lt.0) then
    c(m)=c(m)+10
    c(m+1)=c(m+1)-1
  endif
enddo
endif
endif
endif

```

$|a| - |b|$ を計算して
 $|c|$ を求めていく
 1 桁目から順に $c(m)$ を求めて
 いき、 $c(m) < 0$ のときは
 $c(m)$ に 10 を加えて、
 $c(m+1)$ から 1 を引いて
 繰り返し下りの処理をする

```

write(*,*) , 'hugou=',cc
do m=n,1,-1
  write(*,*) , c(m)
enddo

end

```

計算結果を表示

プログラム終了(以下の部分
 は初期値入力のサブルーチン)

```

subroutine nyuuryku(ashoki,bshoki,ahugou,bhugou,a,b,aa,bb,n,s,t)
integer n,s,t
integer a(n),a1(s),aa,ahugou
integer b(n),b1(t),bb,bhugou,m
character*10 ashoki
character*10 bshoki

aa=ahugou
bb=bhugou

do m=1,s
  a1(s+1-m)=ichar(ashoki(m:m))-48
  a(s+1-m)=a(s+1-m)+a1(s+1-m)
enddo

```

```

do m=1,t
  b1(t+1-m)=ichar(bshoki(m:m))-48
  b(t+1-m)=b(t+1-m)+b1(t+1-m)
enddo

end

```

プログラム終了

なお、減法のプログラムは
 $a-b$ を $a+(-b)$ と考えて、
 上のプログラムにおいて
 a 、 b の初期値を入力した後に
 $bb=-bb$ という一行を
 7 ページ目の * 行部分に
 加えればよい

実行結果(加法の場合)

```

hugou=          -1
              0
              0
              0
              0
              0
              0
              0
              0
              0
              0
              2

```

-678 と 676 の和-2 が求められた

実行結果(減法の場合)

```

hugou=          -1
              0
              0
              0
              0
              0

```

0
1
3
5
4

-678 と 676 の差-1354 が求められた

- 乗法の多倍長演算のプログラム($a \times b = c$ の c を求めるプログラム)

```
integer n,s,t
parameter(n=10)
parameter(s=5)
parameter(t=3)
integer a(n),b(n),c(n),e(n),aa,bb,cc
integer m,r,x
integer a1(s),b1(t),ahugou,bhugou
character*10 ashoki
character*10 bshoki
```

```
ahugou=1
ashoki='12345'
bhugou=-1
bshoki='678'
```

ここでは、 $a=12345$ 、 $b=-678$
として、 $a \times b$ の積 c を求める

```
do m=1,n
  a(m)=0
  b(m)=0
  c(m)=0
enddo
```

```
do m=1,s
  a1(m)=0
enddo
```

```
do m=1,t
  b1(m)=0
```

```
enddo
```

```
call nyuuryku(ashoki,bshoki,ahugou,bhugou,a,b,aa,bb,n,s,t)
```

↑ここまでが初期値入力の部分
以降の行で乗法の演算を
行っていく

```
cc=aa*bb
```

乗法では、積 c の符号は
 a 、 b の符号同士の積で求まる

```
do m=1,n
```

```
  do r=1,n
```

```
    c(m+r-1)=a(m)*b(r)+c(m+r-1)
```

```
    if(c(m+r-1).ge.10) then
```

```
      x=c(m+r-1)/10
```

```
      c(m+r)=c(m+r)+x
```

```
      c(m+r-1)=c(m+r-1)-10*x
```

```
    endif
```

```
  enddo
```

```
enddo
```

```
write(*,*) , 'hugou=',cc
```

```
do m=n,1,-1
```

```
  write(*,*) , 'c=',c(m)
```

```
enddo
```

まず $a(1)$ を $b(1) \sim b(n)$ に
順に掛けていき、その積を
 $c(1) \sim c(n)$ の対応する箱に
順次加えていく
それが終わったら $a(2) \sim a(n)$
も同様に $b(1) \sim b(n)$ と掛ける。
もし計算途中で $c(m)$ の箱に
10 以上の数が入った時は、
 $c(m)$ を 10 で割った商 x を求め、
 $c(m)$ から $10 \times x$ を引いて 1 桁の
整数が $c(m)$ に入るようにし、
 $c(m+1)$ に x を加えて
繰り上がりの処理をする

```
end
```

計算結果をリスト表示して
プログラムのメイン部分終了
以下は数値入力のサブルーチン

```
subroutine nyuuryku(ashoki,bshoki,ahugou,bhugou,a,b,aa,bb,n,s,t)
```

```
integer n,s,t
```

```
integer a(n),a1(s),aa,ahugou
```

```
integer b(n),b1(t),bb,bhugou,m
```

```
character*10 ashoki
```

```
character*10 bshoki
```

```

aa=ahugou
bb=bhugou

do m=1,s
  a1(s+1-m)=ichar(ashoki(m:m))-48
  a(s+1-m)=a(s+1-m)+a1(s+1-m)
enddo

do m=1,t
  b1(t+1-m)=ichar(bshoki(m:m))-48
  b(t+1-m)=b(t+1-m)+b1(t+1-m)
enddo

end

```

プログラム終了

実行結果

```

hugou=      -1
c=          0
c=          0
c=          0
c=          8
c=          3
c=          6
c=          9
c=          9
c=          1
c=          0

```

-12345 と 678 の積
-8369910 が求められた

除法のプログラムに関しては、完成させることができなかつたため、ここには掲載していない

・参考文献

Fortran 学習帳 -Fortran の基礎・基本から始める講座 初心者のための文法学習・入門用の初級・簡単プログラミング講座- intel visual コンパイラ インテル 関数
(<http://www.geocities.jp/eyeofeconomyandhealth/index.html>)