Basicの基礎・再帰関数

０．目次

８．再帰関数

８．１　階乗関数

再帰関数は、関数の中で自分自身を呼び出す関数をいう。関数を簡潔に定義することができる。
階乗関数f(n) (n≧0)を明示的に書くとつぎのようになる。

```
f(n) = n*(n-1)*(n-2)・・・2*1  (n≧1)
     = 1                       (n=0 )
```

●再帰的定義

階乗f(n)(n≧0)は、つぎのように再帰的に定義される。

```
f(n) = n * f(n-1)   (n≧1)
     = 1            (n=0 )
```

（注意）
・左辺の関数fが右辺に現れている。
・パラメータnについて、右辺ではn-1になっている。
このため、この定義を繰り返し適用すると、n=0の値1にたどり着く。
その結果、f(n)の値が確定する。

n=4の場合では、次のようになる。

```
f(4) = 4*f(3)
     = 4*3*f(2)
     = 4*3*2*f(1)
     = 4*3*2*1
     = 24
```

階乗関数f(n)は、Function文を使って定義通りに書ける。

●プログラム(K811.bas)

```
1  ' << K811.bas >>
2  ' 階乗関数
3  '
4  ' 正整数Nの読み込み。
5  Read N
6  '
7  ' 階乗関数f(n)の計算。
8  Print"f(";N;")=";F(N)
9  End
10 '
11 ' 関数：階乗関数。
12 Function F(N)
13   If( N = 0 ) Then
```

```
14      Z = 1
15    Else
16      Z=N*F(N-1)
17    End If
18    ' 戻り値Fの設定。
19    F=Z
20  End Function
21  '
22  ' データ。
23  Data 4
```

実行結果

```
f( 4)= 24
OK
```

●n=4 の場合、プログラムK811.basの動作は①から⑨のようになる。

①f(4)はf(3)を呼び出す。 f(4)↓↑f(3)　⑨f(4)は値24を返す。
②f(3)はf(2)を呼び出す。 f(3)↓↑f(2)　⑧f(3)は値6をf(4)に返す。
③f(2)はf(1)を呼び出す。 f(2)↓↑f(1)　⑦f(2)は値2をf(3)に返す。
④f(1)はf(0)を呼び出す。 f(1)↓↑f(0)　⑥f(1)は値1をf(2)に返す。
⑤f(0)は値1をf(1)に返す。

● 階乗関数f(n)の動作解析

プログラム(K812.bas)

```
1   ' << K812.bas >>
2   ' 階乗関数
3   '
4   ' 正整数Nの読み込み。
5   Read N
6   '
7   ' 階乗関数f(n)の計算。
8   RST=F(N)
9   Print"f(";N;")=";RST
10  End
11  '
12  ' 関数：階乗関数。
13  Function F(N)
14    Print"-->";N
15    If( N = 0 ) Then
16      Z = 1
17    Else
```

```
18      Z=N*F(N-1)
19    End If
20    F=Z
21    Print"<-- N=";N;" Z=";Z
22  End Function
23  '
24  ' データ。
25  Data 4
```

実行結果

```
 1  --> 4
 2  --> 3
 3  --> 2
 4  --> 1
 5  --> 0
 6  <-- N= 0 Z= 1
 7  <-- N= 1 Z= 1
 8  <-- N= 2 Z= 2
 9  <-- N= 3 Z= 6
10  <-- N= 4 Z= 24
11  f( 4)= 24
12  OK
```

８．２　基本演算

８．２．１　乗算

m,nの乗算 n×m の再帰的定義は次のようになる。

●再帰的定義

m,nを正整数とする。

```
mul(n,m) = 0                (m=0)
         = n + mul(n,m-1)   (m≧1)
```

●乗算関数mul(n,m)の計算。n=5,m=3の場合。

```
mul(5,3) = 5 + mul(5,2)
         = 5 + 5 + mul(5,1)
         = 5 + 5 + 5 + mul(5,0)
         = 5 + 5 + 5 + 0
         = 15
```

乗算関数mul(n,m)は、定義通りに書ける。

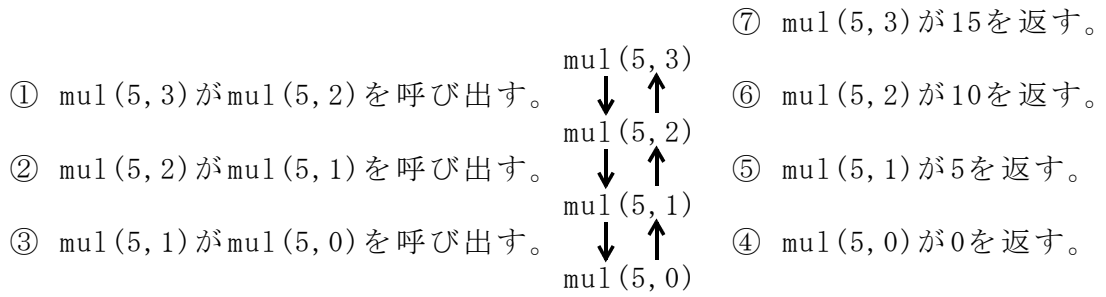
●プログラム(K821.c)

```
1  ' << K821.c >>
2  ' 乗算
3  '
4  Do
5    ' 正整数N,Mの読み込み。
6    Read N,M
7    If (N <= 0) Or (M <= 0) Then Exit Do
8    RST=Mul(N,M)
9    Print"Mul(";N;",";M;")=";RST
10 Loop
11 End
12 '
13 ' 関数：乗算関数。
14 Function Mul(N,M)
15   If( M = 0 ) Then
16     Z=0
17   Else
18     Z=N+Mul(N,M-1)
19   End If
20   Mul=Z
21 End Function
22 '
23 ' データ。
24 Data 2,3,  4,5,  0,0
```

実行結果

```
Mul( 2, 3)= 6
Mul( 4, 5)= 20
OK
```

●プログラムK821.basの動作。n=5,m=3の場合、①から⑦のようになる。



８．２．２　除算

m,nの乗算 n/m の再帰的定義は次のようになる。

●再帰的定義

m,nを正整数とする。nをmで割る。

$$
\begin{aligned}
div(n,m) &= 0 && (n<m)\\
&= 1 + div(n-m,m) && (n \geqq m)
\end{aligned}
$$

●除算関数div(n,m)の計算。n=15,m=4の場合。

$$
\begin{aligned}
div(15,4) &= 1 + div(11,4)\\
&= 1 + 1 + div(7,4)\\
&= 1 + 1 + 1 + div(3,4)\\
&= 3
\end{aligned}
$$

除算関数div(n,m)は、定義通りに書ける。

●プログラム(K822.bas)

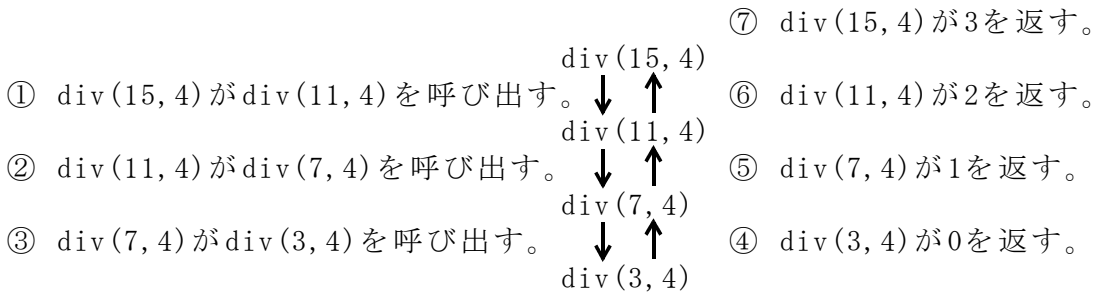
```
1 ' << K822.bas >>
2 ' 除算
3 '
4 Do
5   ' 正整数N,Mの読み込み。
6   Read N,M
7   If (N <= 0) Or (M <= 0) Then Exit Do
8   RST=Div(N,M)
9   Print"Div(";N;",";M;")=";RST
```

```
10  Loop
11  End
12  '
13  ' 関数：除算関数。
14  Function Div(N,M)
15    If N < M Then
16      Z=0
17    Else
18      Z=1+Div(N-M,M)
19    End If
20    Div=Z
21  End Function
22  '
23  ' データ。
24  Data 12,3,  12,4,  12,5,  0,0
```

実行結果

```
Div( 12,  3)= 4
Div( 12,  4)= 3
Div( 12,  5)= 2
OK
```

●プログラムK822.basの動作。n=15,m=4の場合、①から⑦のようになる。



８．３　フィボナッチ関数

フィボナッチ数について考察する。

●再帰的定義

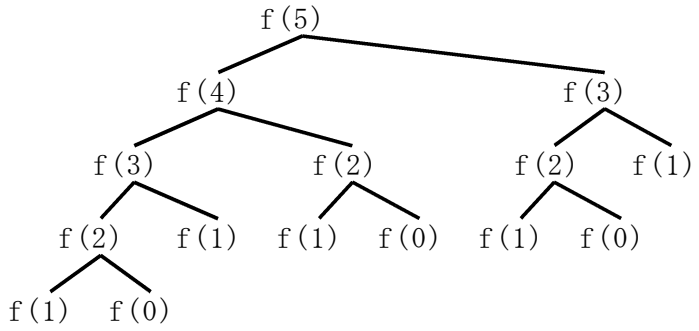
フィボナッチ関数はつぎのように定義される。

```
f(n) = f(n-1) + f(n-2)   (n≧2)
     = 1                 (n=1 )
     = 1                 (n=0 )
```

●フィボナッチ関数f(n)の計算。n=5 の場合。

```
f(5) = f(4) + f(3)
     = f(3) + f(2) + f(2) + f(1)
     = f(3) + 2*f(2) + f(1)
     = f(2) + f(1) + 2*(f(1) + f(0)) + f(1)
     = f(2) + 4*f(1) + 2*f(0)
     = f(1) + f(0) + 4*f(1) + 2*f(0)
     = 5*f(1) + 3*f(0)
     = 8
```

●フィボナッチ関数f(n)の計算状況。n=5 の場合。



（注意）同じ値f(k)が
何度も計算され、
効率が悪くなる。

f(5)	1 回
f(4)	1 回
f(3)	2 回
f(2)	3 回
f(1)	5 回
f(0)	3 回

フィボナッチ関数f(n)は、定義通りに書ける。

●プログラム(K831.bas)

```
1 ' << K831.bas >>
2 ' フィボナッチ数
3 '
4 ' 正整数Nの読み込み。
5 Read N
6 RST=F(N)
7 Print"f(";N;")=";RST
```

```
 8 End
 9 '
10 ' 関数：フィボナッチ関数。
11 Function F(N)
12   If (N = 0) Or (N = 1) Then
13     Z = 1
14   Else
15     Z=F(N-1)+F(N-2)
16   End If
17   F=Z
18 End Function
19 '
20 ' データ。
21 Data 4
```

実行結果

```
f( 4)= 5
OK
```

●フィボナッチ関数f(n)の動作解析

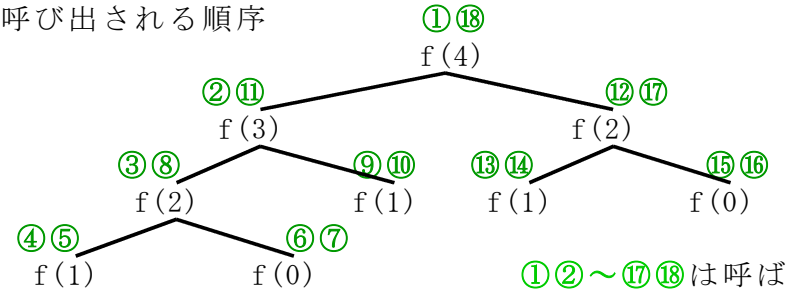
プログラム(K832.bas)

```
 1 ' << K832.bas >>                                    '
 2 ' フィボナッチ数
 3 '
 4 ' 正整数Nの読み込み。
 5 Read N
 6 RST=F(N)
 7 Print"f(";N;")=";RST
 8 End
 9 '
10 ' 関数：フィボナッチ関数。
11 Function F(N)
12   Print"--> N=";N
13   If (N = 0) Or (N = 1) Then
14     Z=1
15   Else
16     Z=F(N-1)+F(N-2)
17   End If
18   F=Z
19   Print"<-- N=";N;" Z=";Z
20 End Function
21 '
22 ' データ。
23 Data 4
```

実行結果

```
①  --> N= 4
②  --> N= 3
③  --> N= 2
④  --> N= 1
⑤  <-- N= 1 Z= 1
⑥  --> N= 0
⑦  <-- N= 0 Z= 1
⑧  <-- N= 2 Z= 2
⑨  --> N= 1
⑩  <-- N= 1 Z= 1
⑪  <-- N= 3 Z= 3
⑫  --> N= 2
⑬  --> N= 1
⑭  <-- N= 1 Z= 1
⑮  --> N= 0
⑯  <-- N= 0 Z= 1
⑰  <-- N= 2 Z= 2
⑱  <-- N= 4 Z= 5
   f( 4)= 5
   OK
```

関数f(n)が呼び出される順序



①②～⑰⑱は呼ばれる順序

大きいnの値について実行時間を測定する。

●プログラム(K833.bas)

```
1  ' << K833.bas >>
2  ' フィボナッチ数
3  '
4  ' 正整数Nの読み込み。
5  Read N
6  '
7  ' 実行時間の測定。
8  T0=Timer: ' 関数Timerは、その日の午前0時からの秒数を
9            ' 返す。
10 RST=F(N)
11 T1=Timer
12 '
```

```
13 Print"f(";N;")=";RST
14 Print"実行時間：";(T1-T0);"秒"
15 End
16 '
17 ' 関数：フィボナッチ関数。
18 Function F(N)
19   If (N = 0) Or (N = 1) Then
20     Z=1
21   Else
22     Z=F(N-1)+F(N-2)
23   End If
24   F=Z
25 End Function
26 '
27 ' データ。
28 Data 30
```

実行時間

```
f( 30)= 1346269
実行時間： 4.61500000000000199秒
OK
```

指定した範囲のf(n)(0≦n≦10)について、計算結果を配列に保存し、２度目以降計算をしないようにして、プログラム（K833.bas）を改良する。

●プログラム(K833a.bas)

```
 1 ' << K833a.bas >>
 2 ' フィボナッチ数
 3 '
 4 Public G(10): ' 配列Gは、Public変数として宣言される。
 5               ' Public変数はすべてのプログラムの中で,
 6               ' 共通に使える変数のことをいう。
 7 ' 正整数Nの読み込み。
 8 Read N
 9 '
10 ' 初期設定。
11 G(0)=1: G(1)=1
12 For I=2 To 10
13   G(I)=G(I-1)+G(I-2)
14 Next I
15 '
16 ' 実行時間の測定。
17 T0=Timer: ' 関数Timerは、その日の午前０時からの秒数を
18           ' 返す。
19 RST=F(N)
20 T1=Timer
21 '
22 Print"f(";N;")=";RST
```

```
23 Print"実行時間：";(T1-T0);"秒"
24 End
25 '
26 ' 関数：フィボナッチ関数。
27 Function F(N)
28   ' 配列Gは、Public変数として宣言されているので、
29   ' Functionブロック内でも使える。
30   If N <= 10 Then
31     Z=G(N)
32   Else
33     Z=F(N-1)+F(N-2)
34   End If
35   F=Z
36 End Function
37 '
38 ' データ。
39 Data 30
```

実行時間

```
f( 30)= 1346269
実行時間：  0.061999999999976126秒
OK
```

８．４　課題

課題１　２項係数

２項係数を再帰的定義にしたがって求めよ。

●再帰的定義

２項係数 c(n,r) は、つぎのように、定義される。

$$
\begin{aligned}
c(n,r) &= c(n-1,r) + c(n-1,r-1) && (n \geqq 2, 1 \leqq r \leqq n-1)\\
&= 1 && (n \geqq 0,\ r=0\ \ )\\
&= 1 && (n \geqq 1,\ r=n\ \ )
\end{aligned}
$$

c(n,r)	0	1	2	3	4	5
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	6	4	1	
5	1	5	10	10	5	1

２項係数c(n,r)は、定義通りに書ける。

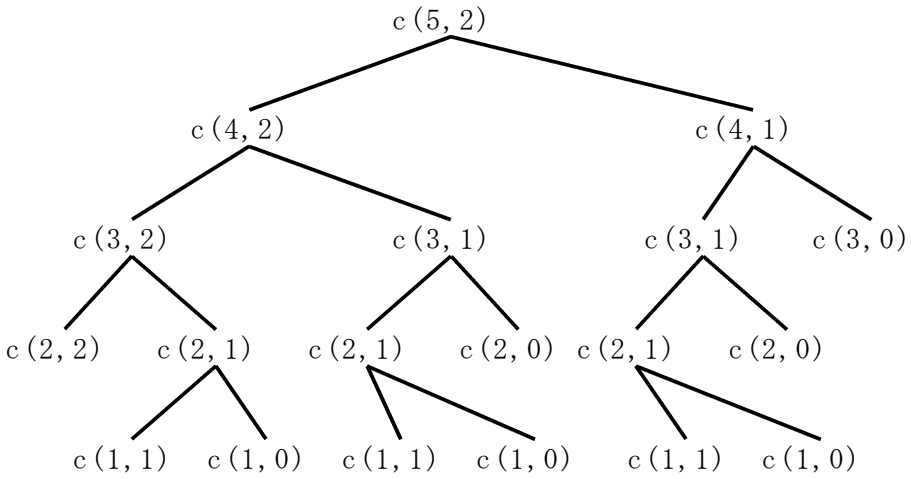
●プログラム(K841.bas)

```
1  ' << K841.bas >> */
2  '  ２項係数の計算
3  Do
4    '  正整数N,Rの読み込み。
5    Read N,R
6    If (N < 0) Or (R < 0) Or (N < R) Then Exit Do
7    RST=C(N,R)
8    Print"c(";N;",";R;")=";RST
9  Loop
10 End
11 '
12 '  関数：２項係数の計算。
13 Function C(N,R)
14   If (R = 0) Or (N = R) Then
15     Z=1
16   Else
17     Z=C(N-1,R)+C(N-1,R-1)
18   End If
19   C=Z
20 End Function
21 '
22 '  データ。
23 Data 3,2,  10,5,  -1,-1
```

実行結果

```
c( 3,  2)= 3
c( 10,  5)= 252
OK
```

● 2項係数c(n,r)の計算過程。n=5,r=2の場合。



（注意）同じ計算を繰り返すので大きなn,rに対して効率が悪くなる。