応用数学　　1999.4.22

２．基本的入出力と分岐 【F文とSELECT文】　　飯島

２－０　はじめに

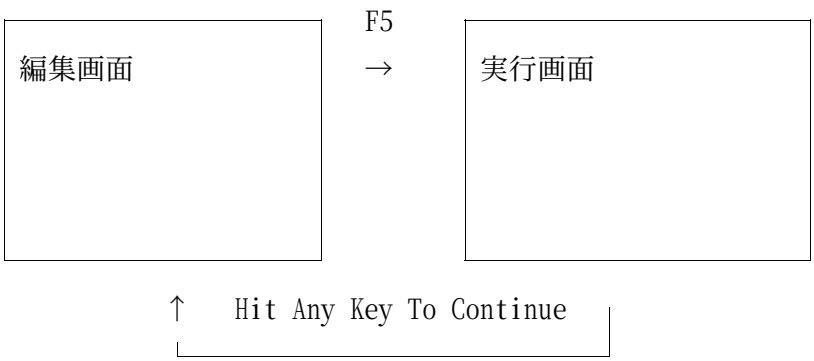
今回は二回目。「まだ二回目」とも言えますが，二回目だって，プログラミング言語にピッタリあっていることなら，数学的なことを調べることもできます。前回の不具合は修正できたはずです。また，ついでに，「key-game.bas」の使い勝手を少しよくし，記録を見ることができるように修正したものを「key-1.bas 」として，追加しておきました。まずは，それで練習してみましょう。基本的な機能はあまり変わりませんが，記録が見やすくなったというだけでも，結構「競う気になる」のではないかと思っています。

さて，前回の残りを終えたら，整数のある話題に注目することにします。

また，前回の「ＫＥＹ－ＧＡＭＥ．ＢＡＳ」の「単語編」が「ＫＥＹ－２．ＢＡＳ」です。ここで入力する単語は，ＱＢＡＳＩＣのキーワードです。

まず，これを使って，指ならしをしてみましょう。今後も，折りにふれてトレーニングをしてみてください。Key-2.bas の方は，以前のままにしてあって，key-1.bas のような改良はしていません。もし，このような改良に関心があって，現在でもそれなりにプログラミング能力のある方は，末尾にkey-1.bas のリストを添付しますので，それを参考に改良してみてください。（学年末の頃には，そのような作業を誰もができるところまで持っていきたいものだと思っています。）

先週および，今の作業から推察できると思いますが，QBでは，次の二つの世界の使い分けをしています。



プログラムの入力および，編集作業をしてみて，これでいいかなと思ったら，それを「実行」してみる。そして，その結果を元にして，元の編集画面の中で再編集をする。

そういう作業の繰り返しです。

編集用のキーとして，次のものをよく使います。

Delete	カーソルの部分の文字を消去する。
BackSpace	カーソルの前の部分の文字を消去する。
Enter	その部分を改行する。冒頭であれば，新しい行を加える。
Insert	挿入モードと上書モードの切り替え

また，先週は FOR　文を使いましたが，リファレンスマニュアルやテキストでこれを調べてみると，どんな記述があるでしょう。そのような記述を読んだり，実際にサンプルプログラムを入力し，実行しながら，その意味を理解していくというのことが，プログラミングを覚えていくための一つの方法です。

また，授業中などに，QBASICを使っているときに，調べたいと思うのも不思議なことではありません。そういうときには，該当する文字のところに，カーソルを合わせて（点滅する場所を合わせて），F1 キーを押してみましょう。すると，FOR に関する画面が出てきます。英語なので，使いにくいのは確かかもしれませんが，こういう形でマニュアルを参照することも可能です。

ちなみに，このようなマニュアルをオンラインマニュアルとか，オンラインヘルプと言います。また，日本語版のQuick BASIC をはじめ，各種の言語では，日本語でそのようなものが表示されるようになっています。

２－１　「入出力」という概念　－　INPUT と PRINT　－

プログラムあるいは，コンピュータというものは，勝手になにかをしてくれるものではありません。命令があれば，それを忠実に実行するというだけの存在です。そういうものとしてのプログラムの最も基本的な構成は，

入力→処理→出力

です。何が入力であり，何が出力なのかを理解すること，またそれを考えて設計することが最も基本的です。

もちろん，中には「入力がない」プログラムもあります。例えば，

```
FOR i = 1 to 10
    PRINT i;
NEXT
```

というプログラムでは，利用者が入力すべきものはありません。しかし，多くの場合，そういうプログラムは，あまり価値がありません。あるいは，プログラムの一部を変えながら使うことを想定しているものが多いと言ってもいいでしょう。例えば，このプログラムの場合，入力を明確化すると，次のようなプログラムになります。

```
INPUT LastNumber
FOR i = 1 to LastNumber
    PRINT i;
NEXT
```

このように，プログラム自身の中では明確になっていないとしても，何が入力であり，何が出力になっているのかを考えていく習慣をつけましょう。

２－２　代入としての式　－　総和を求める問題

入力から出力を作る「処理」において，最も基本的な役割を果たすのは様々な「式」です。それは，数学での計算が果たす役割と一緒です。しかし，数学での「式」の意味と，プログラムにおける「式」の意味は異なる部分がかなりあります。例えば，次の式を見てみましょう。

$$x = x + 1$$

これは数学的には「解なしの方程式」でしかありません。いわばナンセンスです。しかし，プログラムの中ではよく使われます。この意味は，

「ｘの値に１を加えたものを新たにｘの値とする」

ということです。よりコンピュータに密接に述べると，

「変数ｘという番地に記憶されている数値に１を加え，これを変数ｘの番地に書き込め」

ということです。つまり，＝の役割は，「等式」としての役割ではなく，「代入」ということです。また，ここで注意しておきたいのは，

「プログラムにおける『変数』というのは，数値が記憶される『箱』を指す」

ということです。

このことを念頭に置いて，次のプログラムの意味を考えてみましょう。

```
()
FOR i = 1 to 100
     sum = sum + i
NEXT
PRINT sum
```

(2)

```
FOR i = 1 to 100
      sum = sum + (2 * i + 1)
NEXT
PRINT sum
```

(3)

```
FOR i = 1 to 201 STEP 2
      sum = sum + i
NEXT
PRINT sum
```

練習問題

(1) N を入力し，1 からN までの和を求めるプログラムを作れ。
その結果を下に表にせよ。

N	1	2	3	4	5	6	7	8	9	10	20	30	40	50
N までのk の和														

(2) 1 からN までの和を求めるとき，初めて20000 を越えるのはN がいくつのときか

(3) 1 からN 個の奇数の和を求めるプログラムを作れ。
いろいろと調べて気がつくことはないか。

(4) 1 からN 個のk^2 の和を求めよ。
その結果を下に表にせよ。

N	1	2	3	4	5	6	7	8	9	10	20	30	40	50
N までのk^2 の和														

(5) 1 からN 個のk^3 の和を求めよ。その結果を下に表にせよ。

N	1	2	3	4	5	6	7	8	9	10	20	30	40	50
N までのk^3 の和														

何か気がつくことはあるか。

2－3　条件の判定としての式　－　IF THEN ～　と IF THEN　～ ELSE ～　－

もう一つの「式」の役割は，条件の明示とする判定である。そして，多くの場合，それはIF文との組み合わせで使われることが多い。例えば，

(1)

```
DO
    INPUT n
    IF n > 10 THEN PRINT n; ”is big number.”
LOOP until n = 0
```

(2)

```
DO
    INPUT n
    IF n > 10 THEN PRINT n; ”is big number.”ELSE PRINT n; ”is small.”
LOOP until n = 0
```

２－４　IF　ブロック

　また，プログラムを見やすくしたい場合や，処理する内容が多い場合には，次のような IF　ブロックを使います。

例(3)

```
DO
    INPUT n
    IF n > 10 THEN
        PRINT n; ”is big number.”
    ELSE
        PRINT n; ”is small.”
    END IF
LOOP until n = 0
```

　ここで特徴的なのは，IFから始まるブロックの最後を ENDIF　で明示する部分ですが，このようなEND***という使い方は他でもよく使います。

２－５　これは何でしょう。

次のプログラムは何をするプログラムでしょう。

例(4)

```
INPUT n
FOR i = 2 to n-1
    IF (n MOD i) = 0 THEN
        check = -1
        EXIT FOR
    END IF
NEXT
IF check = -1 THEN
    PRINT n ; ”is not prime number.”
ELSE
    PRINT n ; ”is a prime number.”
END IF
```

例(5)　例(4)を少し修正すると，

```
DO
    INPUT n
    check = 0
    IF n = 0 then EXIT DO
    FOR i = 2 to n-1
```

```
        IF (n MOD i) = 0 THEN
            check = -1
            EXIT FOR
        END IF
    NEXT
    IF check = -1 THEN
        PRINT n ; "is not prime number."
    ELSE
        PRINT n ; "is a prime number."
    END IF
LOOP
```

になります。これを使って，できるだけ多くの，そしてできるだけ大きなprime numberを見つけましょう。

prime number

(3) (2)を更に修正したいと思います。どんな観点から，どのように修正することができるでしょうか。また，これらを使うと，どんなことを調べることでできるでしょうか。

２－６　SELECT　文

IF文は，一つの条件を元に，二つの分岐をしますが，一つの式の結果から，いろいろな分岐をする場合には，SELECT　文を使う方が適してきます。

(1) 例

```
INPUT n
FOR i = 1 to n
    IF n MOD i = 0 then Yakusuu = Yakusuu + 1
NEXT
SELECT CASE Yakusuu
    CASE 1
        PRINT "n = 1, isn't it?"
    CASE 2
        PRINT n;"is a prime number"
    CASE ELSE
        PRINT n;"is not a prime number"
END SELECT
```

SELECT　文にはもう少し応用した使い方もありますが，必要なときに紹介することにし，ここでは上記の最も簡単な使い方を理解しておいてください。

２－７　問題

(1) n を入力し，n の約数をすべて表示するプログラムを作れ。

(2) 約数の個数が３個になる数を５つ見つけよ。

(3) 約数の個数が奇数個になる数と偶数個になる数にはどういう違いがあるか。

(参考資料 : key-1.basのリスト)

```
DECLARE SUB PrintScores ()
DECLARE SUB KeyGame ()
DECLARE FUNCTION inkey2$ ()
DECLARE SUB PrintHighScores ()
DECLARE SUB PrintHighScore ()
DECLARE FUNCTION RealTime! ()

  TYPE keyData
    D AS STRING * 10
    T AS STRING * 8
    RT AS SINGLE
```

```
  miss AS SINGLE
  score AS SINGLE
END TYPE


COMMON SHARED Min AS SINGLE, LogFile$, OldTime AS SINGLE
LogFile$ = "Key-Game.His"
OPEN LogFile$ FOR APPEND AS #1
CLOSE #1

CLS
RANDOMIZE TIMER
COLOR 7
DO
  LOCATE 1, 1
  PRINT "Keyboard Trainer  (.Iijima)";
  CALL PrintHighScore
  COLOR 3
  PRINT "           Space";
  COLOR 7
  PRINT ": Start , ";
  COLOR 3
  PRINT "R";
  COLOR 7
  PRINT ": Ranking (Top 20), ";
  COLOR 3
  PRINT "L";
  COLOR 7
  PRINT ": List of Data, ";
  COLOR 3
  PRINT "Q";
  COLOR 7
  PRINT ": End"
  COLOR 7
  a$ = UCASE$ (nkey2$)
  CLS
  OldTime = 0
  LOCATE 3, 1
  SELECT CASE a$
  CASE " ": CALL KeyGame
```

```
    CASE "R": PrintHighScores
    CASE "L": PrintScores
    CASE "Q": END
    END SELECT
  LOOP

FUNCTION inkey2$
  DO
     a$ = INKEY$
  LOOP UNTIL a$ <> ""
  inkey2$ = a$
END FUNCTION

SUB KeyGame
  dummy = RealTime
  FOR i = 1 TO 20
    a$ = CHR$ (((ASC ("Z") - ASC ("A")) * RND) + ASC ("A"))
    PRINT RIGHT$ ("       " + STR$ (i), 7); ": "; a$; "  ";
    b$ = ""
    DO
      b$ = UCASE$ (INPUT$ (1))
      IF a$ <> b$ THEN
        BEEP
        miss = miss + 1
        PRINT b$;
      ELSE
        PRINT "ok! ";
        PRINT USING "##.##"; RealTime - LastTime;
        PRINT "sec."
        LastTime = RealTime
      END IF
    LOOP UNTIL a$ = b$
  NEXT
  COLOR 2
  PRINT "  Time : "; RealTime;
  PRINT "Miss : "; miss;
  PRINT "   Score : "; RealTime + miss
  COLOR 7
  IF RealTime + miss < Min THEN
     COLOR 4
```

```
      PRINT "You make NEW HIGH SCORE !"
      COLOR 7
    END IF
    OPEN LogFile$ FOR APPEND AS #1
    WRITE #1, DATE$, TIME$, RealTime, miss
    CLOSE #1

END SUB

SUB PrintHighScore
    Min = 1000
    OPEN LogFile$ FOR INPUT AS #1
    DO UNTIL EOF(1)
      INPUT #1, D$, T$, RTime, miss
      IF Min > (RTime + miss) THEN
       MinD$ = D$
       MinT$ = T$
       MinRTime = RTime
       Minmiss = miss
       Min = MinRTime + Minmiss
      END IF
    LOOP
    CLOSE #1
    COLOR 2
    PRINT "    High Score : "; Min
    COLOR 7
END SUB

SUB PrintHighScores

    OPEN LogFile$ FOR INPUT AS #1

    ct = 0
    DO UNTIL EOF(1)
      INPUT #1, D$, T$, RTime, miss
      ct = ct + 1
    LOOP
    DIM x(ct) AS keyData
    SEEK 1, 1
    FOR i = 1 TO ct
```

```
    INPUT #1, x().D, x().T, x().RT, x().miss
    x().score = x().RT + x().miss
    'PRINT i; x().D; x().T; x().RT; x().miss, x().score
  NEXT
  CLOSE #1

  FOR i = 1 TO ct
    FOR j = 1 TO ct
    IF i > j AND x().score < x(j).score THEN SWAP x(), x(j)
    IF i < j AND x().score > x(j).score THEN SWAP x(), x(j)
    NEXT
  NEXT

  PRINT "Top 20 Scores"
  FOR i = 1 TO ct
    IF i > 20 THEN EXIT FOR
    PRINT RIGHT$(" " + STR$(i), 2); ": "; x(i).score; TAB(5); x(i).D; " ";
 x(i).T; x(i).RT; x(i).miss
  NEXT

END SUB

SUB PrintScores

  OPEN LogFile$ FOR INPUT AS #1

  ct = 0
  DO UNTIL EOF(1)
    INPUT #1, D$, T$, RTime, miss
    ct = ct + 1
  LOOP
  DIM x(ct) AS keyData
  SEEK 1, 1
  FOR i = 1 TO ct
    INPUT #1, x(i).D, x(i).T, x(i).RT, x(i).miss
    x(i).score = x(i).RT + x(i).miss
    PRINT i; x(i).D; " "; x(i).T; x(i).RT; x(i).miss, x(i).score
    IF i MOD 20 = 0 THEN
    COLOR 7 + 16
    PRINT "Hit Any Key"
```

```
    COLOR 7
    dummy$ = inkey2$
    END IF
  NEXT
  CLOSE #1
END SUB

FUNCTION RealTime
  IF OldTime = 0 THEN OldTime = TIMER
  RealTime = TIMER - OldTime
END FUNCTION
```