

# プログラミング 及び演習

福山平成大学 経営情報学科

福井正康

## 序文

最近、パソコンの世界で Visual Basic ( V B ) が脚光を浴びています。これは Windows 上で、ビジュアルに、組み立てるようにプログラムを作成する統合環境で、非常に有力なプログラミングツールであると思います。ただ、動作環境が Windows でするので、プログラミングの解説書ではコントロール等の使用法が大部分を占めており、一般的なプログラム学習用としてはかなり特殊な感じがします。そこで、この講義では他の言語にも比較的近く、V B にも移行が楽な Quick Basic ( Q B ) を使用言語に選びました。これですと、COBOL や C、C++ 等への一応の入門用にもなるでしょう。

パソコンではアプリケーションが発達し、次第にそれらのマクロが言語の代わりをするようになってきています。こういった環境はプログラミングの作業自体を変革しています。経済、経営系学部出身者が旧来のプログラミング作業に従事することはますます少なくなっていくでしょう。しかし、V B が各種のアプリケーションの橋渡しをする共通マクロ言語になる可能性も見えてきた今、プログラミング入門用として Q B というのはまんざら悪い選択でもなさそうです。ただ、Q B にはポインタの概念がない等のために、これだけではプログラミングの基礎として完全とはいえません。やはり、C 等の汎用性のある言語をきちんと勉強して、Windows プログラミングに進まれることをお勧めします。

経営情報学科に通っていると自然にプログラムが出来るようになって入学してくる人がたくさんいますが、プログラミングは完全に論理的な作業なので、現在の入学制度では、適性のある人は少数です。そのような人には十分にその才能を伸ばしていただきたいのですが、一般的に経営情報学科の学生はプログラムよりむしろ、コンピュータのユーザーとして、経営データの取り扱いを十分に勉強すべきではないかと思います。プログラミングの教員がこのようなことを言うのは変でしょうか。

福山平成大学 経営学部 経営情報学科

福井正康

# 1章 BASIC入門

## 必要最小限お絵描き命令

命令 CLS  
機能 画面の表示を消す

命令 PSET(X,Y),色  
機能 画面に点を打つ

例 PSET(560,375),4

備考 色 0:黒 1:青 2:緑 3:水 4:赤 5:紫 6:黄 7:白  
(薄い)8:灰 9:青 10:緑 11:水 12:赤 13:紫 14:黄 15:白  
グラフィック画面の座標 図1-1参照

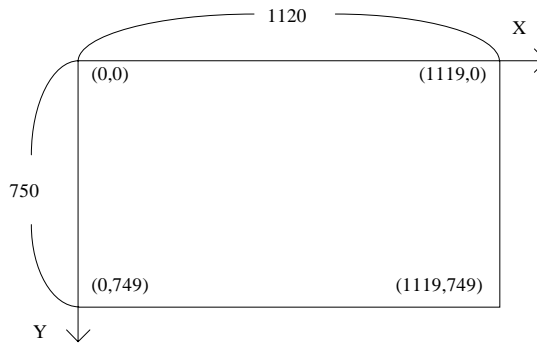


図1-1

命令 LINE(X1,Y1)-(X2,Y2),色[,B OR BF]  
機能 直線を引く、四角形を描く、四角形を描いて中を塗る  
例 LINE(200,200)-(400,400),4 赤の直線  
LINE(400,400)-(600,600),2,B 緑の四角形  
LINE(600,200)-(800,400),1,BF 青の四角形中塗り  
LINE(800,400)-(1000,600),9,BF 薄い青の四角形中塗り

命令 CIRCLE(X,Y),半径,色[, [開始位置],[終了位置],[縦横比]]  
機能 円を描く、楕円を描く、円弧を描く  
例 PI=3.14159

CIRCLE(200,200),100,2	緑の円
CIRCLE(200,400),100,4,0,PI	赤の半円(上半分)
CIRCLE(200,600),100,4,PI,0	赤の半円(下半分)
CIRCLE(400,200),100,4,0.5*PI,1.5*PI	赤の半円(左半分)
CIRCLE(600,200),100,1,,2	青の縦長楕円(縦半径:100)
CIRCLE(600,600),100,1,,0.5	青の縦長楕円(横半径:100)
CIRCLE(800,200),100,6,0,PI,0.5	黄の横長半楕円(上半分)

備考 (X,Y)は円の中心  
半径は長半径  
開始・終了位置は  $-2*3.14159$  から  $2*3.14159$  (ラジアン) で時計と逆回り  
開始・終了位置に - (マイナス) が付くと半径軸も表示される  
縦横比は 縦半径 / 横半径

命令 PAINT(X,Y),色[,境界色]

機能 境界色で区切られた領域を塗りつぶす

例 CIRCLE(400,400),200,2

CIRCLE(600,400),200,2

PAINT(500,400),4,2 緑の境界の領域を赤で塗りつぶす 始点(500,400)

## 参考

時間待ち

```
FOR T=1 TO 50000: NEXT T
```

星空を描く

```
X1=0: Y1=0: X2=1119: Y2=600
```

```
FOR I=1 TO 300
```

```
  X=X1+(X2-X1)*RND
```

```
  Y=Y1+(Y2-Y1)*RND
```

```
  PSET(X,Y),7
```

```
NEXT I
```

キー操作

PF5(実行) , ALT(メニュー) , CTRL+C (プログラムの停止) ,

SHIFT + PF5 (エラーで緑色の中断が起こった場合の再実行)

SHIFT + (範囲指定) , SHIFT + 削除(切り取り) , SHIFT + 挿入(復活)

## 演習 1

- 1) 画面の中央に横 600 ドット、縦 400 ドットの赤い長方形の枠を描け。
- 2) 1) の長方形内部に図 1 - 2 のように赤い楕円の弧を描け。
- 3) 図 1 - 2 の図形内の区切られた部分を図 1 - 3 のように 5 色に塗り分けよ。

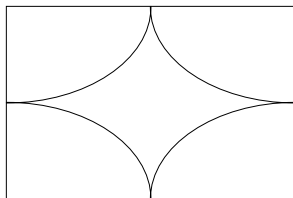


図 1 - 2

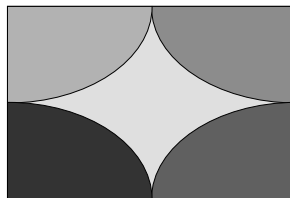


図 1 - 3

- 4) この章で述べた命令を用いて「四季」という題目で絵を描け。

## 2章 データの出力

### 出力命令

命令 PRINT(LPRINT) 変数[文字列]ならび  
機能 ディスプレイ（プリンタ）に変数[文字列]を表示する  
例 PRINT 100  
PRINT "BASIC"  
PRINT "I am ";19;"years old."  
備考 , で区切れれば標準の間隔を開けて表示する。  
; で区切れれば連続して表示する。

### 例文

#### RE12-1 PRINT 文と改行

```
'RE12-1  
PRINT "私は"  
PRINT 19  
PRINT "才です。"  
PRINT "私は";  
PRINT 19;  
PRINT "才です。"
```

#### RE12-2 数値変数

```
'RE12-2  
X=19  
PRINT "私は";X;"才です。"
```

#### RE12-3 文字列変数

```
'RE12-3  
AGE=19  
P$="私は"  
Q$="才です。"  
PRINT P$;AGE;Q$
```

'AGE : 数値変数 P\$,Q\$ : 文字列変数 (後ろに \$ を付ける)

#### RE12-4 四則演算

```
'RE12-4
```

```
X=2: Y=3
P=X+Y: Q=X-Y: R=X*Y: S=X/Y: T=X^Y
PRINT "X=";X
PRINT "Y=";Y
PRINT "X+Y=";P
PRINT "X-Y=";Q
PRINT "X*Y=";R
PRINT "X/Y=";S
PRINT "X^Y=";T
```

'計算の優先順位：() が最優先、次は ^、次は \*/、最後が +-  
'同じ順位の場合は左から計算

注意 例文はそのまま打ち込むと続けて表示されてしまうので、各例文の間に例えば  
INPUT A : CLS を入れる。これによってリターンで画面が消去され次の例に進  
める。

## 演習 2

- 1) 自分の氏名を表示するプログラムを作れ。
- 2) 数値 1 から 5 までを改行しながら表示するプログラムを作れ。
- 3) 5 つの PRINT 文を使って、数値 1 から 5 までを横に表示するプログラムを作れ。
- 4) 数値変数 X に 3、Y に -5 を代入し、X+Y の値を表示するプログラムを作れ。
- 5) 数値変数 X, Y に 3 を代入し、 $X^2+(Y+1)^2$  の値を表示するプログラムを作れ。
- 6) 文字列変数 A\$ に文字列 "福山平成大学" を代入し、それを表示するプログラムを作れ。
- 7) 自分の氏名を文字列変数に代入し、「私は××××です。」と表示するプログラムを作れ。
- 8) 数値変数 X に 2、Y に 3、Z に X+Y を代入し、X, Y, Z を用いて「2 + 3 = 5」と表示するプログラムを作れ。

### 3章 データの入力

#### 入力命令

命令 INPUT ["メッセージ"],変数ならび  
          ["メッセージ"];変数ならび

機能 キーボードから変数へのデータの入力

例 INPUT X  
      INPUT X,Y  
      INPUT "How old are you? ",AGE

備考 メッセージは入力を促す言葉  
      , の場合は何も表示されず続いて入力出来る。  
      ; の場合は ? が表示され、1桁空く。  
      2 変数以上の入力の場合は , で区切って入力する

#### 例文

##### REI3-1 数字の入力

```
INPUT "おいくつですか? ",X           'X は数値変数
PRINT "あなたは";X;"才ですね。"
```

##### REI3-2 文字列の入力

```
INPUT "名前を入力して下さい > ",N$
PRINT "あなたは";N$;"さんですね。"   'N$ は文字列変数
```

##### REI3-3 数字の入力と計算

```
INPUT "X=",X
PRINT "X=";X;"のとき"
PRINT "X^2+X+1="; X^2+X+1
```

##### REI3-4 肥満度の計算

```
PRINT "肥満度の計算"
INPUT "身長(cm) > ",HEIGHT
INPUT "体重(kg) > ",WEIGHT
HYOJUN=(HEIGHT-100.0)*0.9
HIMANDO=WEIGHT/HYOJUN
PRINT
PRINT "肥満度 >>";HIMANDO
```

### 演習 3

- 1) キーボードから変数  $X$  と  $Y$  に数値を入力すると、 $X+Y$ ,  $X-Y$  の値を表示するプログラムを作れ。
- 2) キーボードから変数  $X$  に数値を入力し、 $X$  の 2 乗、 $X$  の 3 乗、 $X$  の  $1/2$  乗 (平方根) の値を表示するプログラムを作れ。但し、 $X^p$  は  $X^{\wedge}p$  と表される。
- 3) 生まれた年 (西暦) をキーボードから入力すると、「今年あなたは 才になりますね。」と今年の誕生日でなる年齢を表示するプログラムを作れ。
- 4) 以下の動きをするプログラムを作れ。但し、 $\times \times$  同士には同じ語句や値が入るものとする。

あなたのお名前は? <名前の入力  $\times \times$ >  
さんの年齢は? <年齢の入力  $\times \times$ >  
さんは  $\times \times$  才ですね。
- 5) 英語、数学、国語、理科、社会の 5 科目の点数をキーボードから入力し、その平均点を表示するプログラムを作れ。



## 4章 データの見やすい出力

### 書式付き出力命令と文字位置・色

命令 PRINT(LPRINT) USING 書式;式[,式・・][;]

機能 データの書式を指定して出力する

例 PRINT USING "###.####";X;  
PRINT USING "& &";"blue";"red";"green"

命令 LOCATE [行][,桁]

機能 ある定まった位置からテキスト表示するために、カーソルの移動を行う。

例 LOCATE 12,38 : PRINT "DATA"

備考 画面の左上隅が (1,1) 右下隅が (25,80)

命令 COLOR [色番号]

機能 出力文字の色を指定

例 COLOR 5

備考 色番号 1:青 2:緑 3:水 4:赤 5:紫 6:黄 7:白

命令 CLS [{0,1,2}]

機能 画面の消去

例 CLS 2

備考 0:テキストとグラフィック, 1:グラフィックのみ, 2:テキストのみ, 指定しない場合は 0 とみなされる

### 例文

#### REI4-1 基本的な書式例

```
PRINT USING "#####";123
PRINT USING "#####";-123
PRINT USING "DATA =#####";123
PRINT USING "####.##";123.4567
PRINT USING "####.####";123.45
PRINT USING "& &";"ABC";"ABCDE";"ABCDEFG"
```

#### REI4-2 指定位置への色付き出力

```
CLS
LOCATE 12,36: COLOR 4
PRINT "FUKUYAMA"
COLOR 7
```

## 演習 4

- 1 ) 200, 30, 2000 を 6 桁右詰で 1 つずつ改行して表示するプログラムを作れ。
- 2 ) 3.1415, 27.18, 141.421 を小数点の桁をそろえて、1 つずつ改行して出力するプログラムを作れ。
- 3 ) "Japan", "USA", "England" の文字列を 10 桁左詰めで横に並べて出力するプログラムを作れ。
- 4 ) プロフィールを入力し終わったら、画面が消えて、それを色付きできれいにテキスト出力するプログラムを作れ。

## 5章 繰返し・FOR文

### FOR～NEXT文

構文 FOR ループ変数=初期値 TO 最終値 [STEP きざみ値]  
<繰返し処理>

NEXT ループ変数

機能 ループ変数を初期値からきざみ値ずつ変化させて最終値まで、くり返し処理を続ける。

例 FOR T=1 TO 50000: NEXT T '時間待ち  
FOR I=1 TO 10: PRINT "Fukuyama Univ.": NEXT I  
FOR NO=1 TO 10: PRINT NO: NEXT NO

備考 きざみ値を省略した場合はきざみ値は 1 となる

### 例文

#### REI5-1 文字列の繰返し表示

```
FOR I = 1 TO 10
  PRINT "FUKUYAMA"
NEXT I
```

#### REI5-2 数値の連続表示

```
FOR I=1 TO 10
  PRINT I
NEXT I
```

#### REI5-3 きざみ値の利用

```
FOR I=2 TO 10 STEP 2
  PRINT I
NEXT I
```

#### REI5-4 合計

```
TOTAL = 0
FOR I = 1 TO 100
  TOTAL = TOTAL + I
NEXT I
PRINT "TOTAL =";TOTAL
```

## REI5-5 2重ループ(九九の表)

```
PRINT "九九の表"  
FOR I = 1 TO 9  
  FOR J = 1 TO 9  
    PRINT USING "####"; I * J;  
  NEXT J  
PRINT  
NEXT I
```

## 演習 5

- 1) 自分の名字を1桁空白を開けて、5個横に表示するプログラムを作れ。
- 2) 1から10まで横に表示するプログラムを作れ。
- 3) 1から10まで4桁右詰めで横に表示するプログラムを作れ。
- 4) 10から1まで4桁右詰めで横に表示するプログラムを作れ。
- 5) 100以上999以下の数をキーボードから入力し、100からその数まで5桁右詰めで横に表示するプログラムを作れ。
- 6) 2から100まで偶数を4桁右詰めで横に表示するプログラムを作れ。
- 7) 9から99まで奇数を4桁右詰めで横に表示するプログラムを作れ。
- 8) 7以上の数をキーボードから入力し、7からその数までの7の倍数を4桁右詰めで横に表示するプログラムを作れ。
- 9) 1から100まで和を計算し、結果を出力するプログラムを作れ。(5050)
- 10) 10から100まで和を計算し、結果を出力するプログラムを作れ。(5005)
- 11) 2から100までの偶数の和を表示するプログラムを作れ。(2550)
- 12) 3から99までの3の倍数の和を表示するプログラムを作れ。(1683)
- 13) 100以上の偶数をキーボードから入力し、100からその数までの偶数の和を表示するプログラムを作れ。
- 14) -100から300までの和を表示するプログラムを作れ。(40100)
- 15) 1から10までの積を計算し、結果を表示するプログラムを作れ。(3628800)
- 16)  $1+(1/2)+(1/3)+\dots+(1/100)$  の結果を表示するプログラムを作れ。(5.187)
- 17)  $5\times 6+6\times 7+7\times 8+\dots+99\times 100$  の結果を表示するプログラムを作れ。(333260)
- 18)  $1/2+2/3+3/4+\dots+10/11$  の結果を表示するプログラムを作れ。(7.980)
- 19)  $1\times 2+3+4\times 5+6+\dots+28\times 29+30$  の結果を表示するプログラムを作れ。(71610)

20)  $a_1=1$ ,  $a_n=5 \times a_{n-1}-1$  のとき、 $a_{10}$  の値を表示するプログラムを作れ。(1464844)

21) 前の問題で  $a_1$  から  $a_{10}$  までの和  $S_{10}$  を表示するプログラムを作れ。(1831057)

22) 自分の名字を横に 5 個、縦に 10 行、表のように出力するプログラムを作れ。

```
福井 福井 福井 福井 福井
福井 福井 福井 福井 福井
福井 福井 福井 福井 福井
.....
福井 福井 福井 福井 福井
```

23) 下のような表を出力するプログラムを作れ。

```
1 2 3 4 5 6 7 8 9 10
2 3 4 5 6 7 8 9 10 11
.....
10 11 12 13 14 15 16 17 18 19
```

24) 1 から 100 までの数で下のような表を出力するプログラムを作れ。

```
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
.....
91 92 93 94 95 96 97 98 99 100
```

25)  $1 \times 1 + 1 \times 2 + \dots + 9 \times 8 + 9 \times 9$  の様に九九を全部足し合わせた結果を表示するプログラムを作れ。(2025)

## 6章 条件判断

### 条件判断構文

構文 IF 条件式 THEN 文1: 文2: ... [ELSE 文3: 文4: ...]

機能 条件式が真ならば文1, 文2, ...を実行し、偽ならば文3, 文4, ...を実行する。

備考 1行で書かなくてはならない。これが GOTO 文の乱用を生む要因となった。

構文 IF 条件式 THEN

文1

文2

....

ELSE IF 条件式 THEN

....

ELSE

....

END IF

機能 条件式の真偽によって実行する文を選び分ける

備考 条件判断によって複数行の実行文に分岐することが出来る

比較 IF X>=0 THEN PRINT X

IF X<0 THEN PRINT -X

IF X>=0 THEN

PRINT X

ELSE

PRINT -X

END IF

### 関係演算子と論理演算子

#### 関係演算子

書式 =, <>(≠), <, <= (=), >, >= (⇒)

#### 論理演算子

書式 NOT, AND, OR

例 X>=0, X<100, X<>100, X>=0 AND X<=10, X<0 OR X>100  
X<-10 OR (Y>0 AND Y<20), (X<0 OR X>1) AND (Y<0 OR Y>1)

### 例文

#### REI6-1 数値の判断

INPUT "数値 > ", X

IF X = 0 THEN

PRINT "ゼロ"

```
ELSEIF X > 0 THEN
  PRINT "正"
ELSE
  PRINT "負"
END IF
```

#### REI6-2 論理演算子

```
INPUT "数值 > ", X
IF X>=0 AND X<=10 THEN
  PRINT "範囲内"
ELSE
  PRINT "範囲外"
END IF
```

#### REI6-3 2種類の IF 文

```
INPUT "点数 > ", TEN
PRINT "判定 1 "
IF TEN>=80 THEN PRINT "優"
IF TEN>=70 AND TEN<80 THEN PRINT "良"
IF TEN>=60 AND TEN<70 THEN PRINT "可"
IF TEN<60 THEN PRINT "不可"
PRINT "判定 2 "
IF TEN>=80 THEN
  PRINT "優"
ELSEIF TEN>=70 THEN
  PRINT "良"
ELSEIF TEN>=60 THEN
  PRINT "可"
ELSE
  PRINT "不可"
END IF
```

#### REI6-4 偶数表示

```
FOR I = 1 TO 100
```

```

    IF I MOD 2 = 0 THEN PRINT USING "####"; I;
NEXT I

```

#### REI6-5 最大を求める

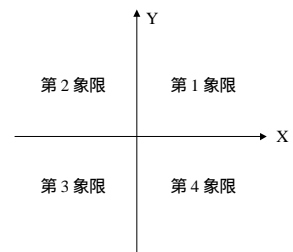
```

INPUT X
MAX = X
FOR I = 2 TO 5
    INPUT X
    IF X > MAX THEN MAX = X
NEXT I
PRINT "最大 ="; MAX

```

### 演習 6

- 1) キーボードから数値を読み込み、その値が 100 以上か 100 未満か判定するプログラムを作れ。
- 2) キーボードから数値を読み込み、その値の絶対値を表示するプログラムを作れ
- 3) キーボードから変数 X と Y に 2 つの数値を読み込んで、その大きさを以下のように比較するプログラムを作れ。例:  $X > Y$ 、 $X = Y$ 、 $X < Y$
- 4) キーボードから変数 X に数値を読み込んで、 $X^2 - X + 1$  の値が領域  $0 < X^2 - X + 1 < 10$  に含まれるかどうか判定するプログラムを作れ。
- 5) (X, Y) の座標をキーボードから入力したら、その点が座標上の第何象限にあるか、判定するプログラムを作れ。ただし、軸上の点は「軸上」と表示すること。
- 6) IF 文を用いて 1 から 100 までの奇数を 4 桁右詰めで表示するプログラムを作れ。
- 7) 5 個の数値をキーボードから読み込み、その最小値を求めるプログラムを作れ。
- 8) 5 科目の得点をキーボードから読み込み、その最高と最低の差を表示するプログラムを作れ。
- 9) キーボードから身長と体重を読み込んで、肥満の状態を示すプログラムを作れ。



ただし、以下の式を用いること。

$$\text{標準体重} = (\text{身長} - 100) \times 0.9$$

$$\text{肥満度} = \text{体重} \div \text{標準体重}$$

$$\text{肥りすぎ} : 1.2 \quad \text{肥満度}$$

$$\text{標準} : 0.8 < \quad \text{肥満度} < 1.2$$

$$\text{やせすぎ} : \quad \text{肥満度} \quad 0.8$$



## 7章 繰返し・while 文

### WHILE 文

```
構文  WHILE 条件式
      文 1
      文 2
      ....
      WEND
```

機能 条件式が満たされる限り、WHILE ~ WEND までを繰返し実行する。

### 例文

#### REI7-1 繰返し条件 1

```
I = 1
WHILE I <= 10
  PRINT USING "####"; I;
  I = I + 1
WEND
```

#### REI7-2 繰返し条件 2

```
I = 1
WHILE I^2 + I + 1 < 100
  PRINT USING "####"; I^2 + I + 1;
  I = I + 1
WEND
```

#### REI7-3 入力条件

```
INPUT "X = ", X
WHILE X < 0
  INPUT "X = ", X
WEND
PRINT "X ^ 0.5 ="; X ^ .5
```

#### REI7-4 回数が決まらない処理

```
INPUT "文字列 > ", A$
WHILE A$ <> ""
```

```

PRINT A$; "が入力されました。"
INPUT "文字列 > ", A$
WEND
PRINT "入力が終わりました。"
REI7-5 回数が決まらない数の合計 (-999 で終了)
TOTAL = 0
INPUT "数値 > ", X
WHILE X <> -999
    TOTAL = TOTAL + X
    INPUT "数値 > ", X
WEND
PRINT "合計 ="; TOTAL

```

## 演習 7

- 1) WHILE 文を使って、2 から 100 までの偶数を 4 桁右詰めで出力するプログラムを作れ。
- 2) キーボードから 0 以上 130 以下の数を読み込み、「あなたは  $x \times x$  ですね。」と答えてくれるプログラムを作れ。但し、これら以外の数が入力された場合は再入力させるようにする。
- 3) キーボードから数値を読み込み、0 が入力されるまで、読み込んだ数値の絶対値を表示し続けるプログラムを作れ。
- 4) キーボードから数値を -999 が入力されるまで読み込み、読み込んだ数値の平均を出力するプログラムを作れ。
- 5) キーボードから整数  $N$  を読み込み、 $N$  自身と 1 を除いた約数のうち最大のものを求めるプログラムを作れ。これら以外に約数がない場合には、「素数」と表示せよ。  
 ヒント：例えば、割る数を  $N-1$  から 1 つずつ小さくして行って、最初に割り切れる整数が求める答えとなる。但し、これが 1 の場合、素数である。
- 6)  $X=0.5$  のとき、 $X^n < 0.001$  となる最小の整数  $n$  を求めるプログラムを作れ。(10)
- 7)  $1+2+3+\dots+n < 500$  となる最大の整数  $n$  とその時の合計を求めるプログラムを作れ。(31,496)

## 8章 算術関数

### 関数書式

関数 ABS(式)

機能 式の絶対値を与える

例 PRINT ABS(-5)

関数 SGN(式)

機能 式の符号 ( + 1 または - 1 ) を与える

例 PRINT SGN(B^2-4\*A\*C)

関数 SQR(式)

機能 式の平方根

例 Y=SQR(2.0)

備考 式が倍精度の場合は倍精度値、その他は単精度値

関数 SIN(式)

機能 式(ラジアン)の正弦

例 Y=SIN(3.1415/6)

備考 式が倍精度の場合は倍精度値、その他は単精度値

関数 COS(式)

機能 式(ラジアン)の余弦

例 X=COS(3.1415/6)

備考 式が倍精度の場合は倍精度値、その他は単精度値

関数 TAN(式)

機能 式(ラジアン)の正接

例 Z=TAN(3.1415/6)

備考 式が倍精度の場合は倍精度値、その他は単精度値

関数 ATN(式)

機能 式の逆正接

例 RAD=ATN(1)

備考 式が倍精度の場合は倍精度値、その他は単精度値  $-\pi/2 < \text{RAD} < \pi/2$  の値

関数 EXP(式)

機能 EXPONENTIAL 関数

例 Y=EXP(-X\*X)

備考 式が倍精度の場合は倍精度値、その他は単精度値

関数 LOG(式)

機能 自然対数値 (  $y = \log_e x$ :  $e = 2.71828$  Napier(ネーピア)の数 )

例 Y=LOG(X)

備考 式が倍精度の場合は倍精度値、その他は単精度値

$\log_{10} x$  の値は公式  $\log_{10} x = \log_e x / \log_e 10$  を用いて  $Y = \text{LOG}(X) / \text{LOG}(10)$  で求まる。

関数 INT(式)

機能 式の値を越えない最大整数値

例 INT(3.14) '3 INT(-3.14) '-4

関数 RND

機能 0 以上 1 未満の少数乱数値

例 X=RND, X=INT(100\*RND)

### 関連命令・関数

コマンド RANDOMIZE 式

機能 式の値に応じた乱数の初期化

例 RANDOMIZE 1

### 例文

#### REI8-1 整数化

```
X1=INT(3.4)
```

```
X2=INT(3.6)
```

```
X3=INT(-3.4)
```

```
PRINT X1; X2; X3
```

#### REI8-2 乱数の発生

```
RANDOMIZE 1
```

```
'0 以上 1 未満の乱数
```

```
FOR I=1 TO 50
```

```
  X=RND
```

```
  PRINT USING "##.##";X;
```

```
NEXT I
```

```
PRINT:PRINT
```

```
'0 以上 100 未満の整数乱数
```

```
FOR I=1 TO 50
```

```
  X=INT(100*RND)
```

```
  PRINT USING "#####";X;
```

```

NEXT I
PRINT:PRINT
'50 以上 200 未満の整数乱数
FOR I=1 TO 50
  X=INT(150*RND+50)
  PRINT USING "#####";X;
NEXT I
PRINT

```

### REI8-3 関数表

```

PI=3.14159
PRINT " DEG";
PRINT " SIN    COS    TAN"
FOR I=0 TO 360 STEP 20
  X=I*PI/180
  PRINT USING "#####";I;
  PRINT USING "###.###";SIN(X);COS(X);TAN(X)
NEXT I

```

### 演習 8

- 1 ) 200 以上 300 未満の整数乱数を 100 個出力するプログラムを作れ。
- 2 ) 1 以上 3 未満の小数乱数を 100 個出力するプログラムを作れ。
- 3 ) (200,200)-(500,400)の 4 角形を描き、その領域内に乱数を用いて星を 200 個描くプログラムを作れ。
- 4 ) (560,300)を中心とした半径 200 の円 (比率 1) を描き、その中に乱数を用いて星を 300 個描くプログラムを作れ。

ヒント : 円を描き、星の個数を最初は 0 とする。

WHILE による判定文 (星の数 < 300)

X は 360-760 の乱数

Y は 100-500 の乱数

もし、円の中心から (X,Y) までの距離が 200 未満なら  
星を描いて、星の数を 1 増やす。

WEND

5) 乱数を用いて半径 1 の円の面積を近似的に求めるプログラムを以下のヒントに従って作れ。(モンテカルロ法)

ヒント: 半径 1 の円の第 1 象限部分で(0,0)-(1,1)の正方形を考える。この正方形の面積は 1 である。この中へ乱数で点を打ち、このうち円の内側へ何割入ったかで円の面積が求まる。すなわち、円の面積 =  $4 \times 1 \times$  割合

6) 以下の値を求めるプログラムを作れ。(2.236, 7.389, 2.303, 0.866, 0.577)

$\sqrt{5}$ ,  $e^2$ ,  $\log_e 10$ ,  $\sin \frac{\pi}{3}$ ,  $\tan 30^\circ$  但し、 $e = 2.71828$  (Napier の数)

7) 0.1 から 0.1 きざみで 2.0 までの  $\log_e x$  と  $\log_{10} x$  の関数表を作成するプログラムを作れ。ただし、 $\log_{10} x = \log_e x / \log_e 10$  の公式を用いよ。

8) 7) を元に  $0.1 \leq x \leq 10.0$  の範囲で、 $\log_{10} x$  のグラフを描くプログラムを作れ。ただし、x,y 軸を引き、出来れば目盛りを入れること。

## 9章 配列の利用

### 配列

非常に多くのデータを扱うときなどは、それぞれのデータに変数を1つずつ与えていたのではプログラムも変数の量も巨大になり、収拾がつかなくなります。そこで、同一名の変数に番号を付けて、管理するとプログラムとしても取扱い易くなり非常に便利です。

### 配列の宣言

書式 DIM 配列名(要素最大値[,要素最大値……])

機能 使用する配列名とその大きさを宣言する

例 DIM A(19) : A(0),A(1),A(2),……,A(19) まで使用可能

DIM STR\$(9) : STR\$(0),STR\$(1),……,STR\$(9)

DIM MAT(3,4) : MAT(0,0),MAT(0,1),……,MAT(3,4)

備考 変数の型は様々にとれる DIM A%(5),B&(5),C!(5),D\$(5),E\$(5)

DIM A(5)      

A(0)	A(1)	A(2)	A(3)	A(4)	A(5)
------	------	------	------	------	------

DIM B(2,3)    

B(0,0)	B(0,1)	B(0,2)	B(0,3)
B(1,0)	B(1,1)	B(1,2)	B(1,3)
B(2,0)	B(2,1)	B(2,2)	B(2,3)

### 例文

#### REI9-1 配列を利用した変数の並べ替え

```
RANDOMIZE 1
DIM A(20)
FOR I = 1 TO 20
  A(I) = INT(10 * RND + 1)
  PRINT USING "###"; A(I);
NEXT I
PRINT
FOR I = 20 TO 1 STEP -1
  PRINT USING "###"; A(I);
NEXT I
PRINT
```

## REI9-2 文字列変数配列の利用

```
DIM CITY$(5)
CITY$(1) = "岡山": CITY$(2) = "倉敷": CITY$(3) = "福山"
CITY$(4) = "尾道": CITY$(5) = "広島"
PRINT "どこに住みたいですか?"
FOR I=1 TO 5
  PRINT I; CITY$(I)
NEXT I
INPUT "番号 > ", N
PRINT "あなたは"; CITY$(N); "市に住みたいのですね。"
```

## REI9-3 2次元配列の転置

```
RANDOMIZE 1
DIM MAT(4, 5)
FOR I = 0 TO 4
  FOR J = 0 TO 5
    K = INT(10 * RND + 1)
    MAT(I, J) = K
    PRINT USING "###"; MAT(I, J);
  NEXT J
  PRINT
NEXT I
PRINT
FOR J = 0 TO 5
  FOR I = 0 TO 4
    PRINT USING "###"; MAT(I, J);
  NEXT I
  PRINT
NEXT J
```

## 演習 9

- 1) 1 以上 100 以下の整数乱数を 100 個配列変数に読み込み、50 未満と 50 以上とに分けて出力するプログラムを作れ



出力例

```

50 未満
 22 44 39 18 5 .....
50 以上
 57 89 72 91 67 .....

```

2) 1 以上 100 以下の整数乱数を 100 個配列変数に読み込み、これを利用して合計を計算して表示するプログラムを作れ。

3) 71, 68, 56, 76, 89, 75, 55, 64, 32, 88 の 10 個の得点を配列に代入し、最高、最低、平均、分散、標準偏差を求めるプログラムを作れ。ただし、分散と標準偏差は平均  $\bar{x}$  を用いて以下の式で与えられる。

$$\text{分散} = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2 \quad \text{標準偏差} = \sqrt{\text{分散}}$$

4) 3) のデータでそれぞれの得点の偏差値を表示するプログラムを作れ。ただし、偏差値は以下の式で与えられる。

$$\text{偏差値} = 50 + 10 * (\text{値} - \text{平均}) / \text{標準偏差}$$

出力例

```

得点      偏差値
 71      ###.#
 68      ###.#
.....
.....
 88      ###.#

```

8) 5 科目の得点を科目名を表示しながら入力し、最高点とその科目名を表示するプログラムを作れ。その際、最高点が 2 科目以上ある場合は考えなくてもよい。

6) 1 以上 100 以下の整数乱数を 10×10 の 2 次元配列に読み込み、縦横の合計を表示するプログラムを作れ。

出力例

```

                                     合計
 11  36  76  55  29  2  45  19  91  24 #####
 33  17  95  5  39  28  63  100  28  44 #####
.....
.....
 76  33  16  88  46  26  51  19  47  24 #####
合計 ##### ##### ##### ##### ##### ##### ##### ##### ##### #####

```

7) 1 以上 100 以下の整数乱数を 100 個配列変数 A に読み込み、以下の方法に従ってデータを小さい順に並べ換えて表示するプログラムを作れ。

方法

1. 変数 N=100 として配列を宣言し、乱数を代入する。
2. ループ 1 : 変数 J は N から 2 まで変化
3.     ループ 2 : 変数 I は 2 から J まで変化
4.          $A(I-1) > A(I)$  ならば値を交換する。
5.     ループ 2 の終わり
6. ループ 1 の終わり
7. データの表示

## ソートについて

数字を大小順に並べ替えたり、文字列をアルファベット順または五十音順に並べ替えることをソートといい、様々な方法が考案されている。上の方法はバブルソートといい、最も簡単だがスピードの遅い例。この他にヒープソート、シェルソート、クイックソート等が有名。例えば 500 個のデータのソート (PC-9801RA/N88BASIC) でも時間に相当の違いがある。

バブルソート	2 分 19 秒	ヒープソート	12 秒
選択ソート	1 分 34 秒	クイックソート	8 秒
シェルソート	11 秒	逆写像ソート	3 秒 等

データの個数が増えれば、バブルソートと選択ソートは殆ど役に立たない。

## 10章 文字列処理

### 文字列の和

```
A$="福山":B$="大学"  
C$=A$+B$  
PRINT C$      '福山大学 と表示
```

### 1 行入力書式

```
命令 LINE INPUT 文字列変数  
機能 1 行分の文字列を入力  
例 LINE INPUT MESS$  
備考 INPUT 文では " " で囲まれた中でしか使えない , (カンマ) も入力  
INPUT A$          入力可   AAA      ABC DEF GHI      "ABC,DEF"  
                  入力不可  ABC,DEF  
LINE INPUT A$    カンマがあっても 1 行丸ごと入力
```

### 文字列処理関数

```
関数 ASC(文字)  
機能 文字の ASCII コードを与える  
例 Z=ASC("A") 'Z=65  
備考 逆は CHR$(式)
```

```
関数 CHR$(式)  
機能 式で与えられる ASCII コードの文字を与える  
例 Z$=CHR$(65) , PRINT CHR$(7)  
備考 逆は ASC(文字)
```

```
関数 INKEY$  
機能 どれかのキーが押されるまでは "" (空)を返し、何か押されると  
その文字列を返す  
例 WHILE INKEY$="" : WEND 'キー入力待ち  
A$=INKEY$ : WHILE A$="" : A$=INKEY$ : WEND '同上  
備考 キー入力待ち , 入力キー判断に利用
```

```
関数 LEN(文字式)  
機能 文字式の文字列の長さを与える  
例 A$="ABCDEFGH"  
PRINT LEN(A$)      '7  
備考 全角文字は 2 文字に数える
```

```
関数 INSTR(式,文字式 1,文字式 2)
```

機能 文字式 1 の<式>番目の文字からの文字式 2 の検索

例 N=INSTR(1, "A:¥TEST.DAT", "DAT")

備考 <文字列 1 >の先頭からの値を返す  
<文字列 2 >が見つからない場合は 0 を返す

関数 LEFT\$(文字式, 式)

機能 文字式の左側の<式>文字分を取り出す

例 A\$=LEFT\$("12:24:30", 2) : HOUR\$=LEFT\$(TIME\$, 2) 'A\$="12"

備考 <式>の値を多く指定し過ぎると文字列の長さで打ち切り

関数 MID\$(文字式, 式 1, 式 2)

機能 文字式の<式 1>番目の文字から<式 2>文字分取り出す

例 A\$=MID\$("12:34:30", 4, 2) 'A\$="34"

関数 RIGHT\$(文字式, 式)

機能 文字式の右側の<式>文字分を取り出す

例 A\$=RIGHT\$("12:34:30", 2) 'A\$="30"

関数 HEX\$(式)

機能 式の数値を 16 進数に変換し、その文字列を与える

例 A\$=HEX\$(27) 'A\$="1B"

関数 OCT\$(式)

機能 式の数値を 8 進数に変換し、その文字列を与える

例 A\$=OCT\$(27) 'A\$="33"

関数 VAL(文字式)

機能 文字式の文字列を数値に直す

例 X=VAL("5"), X=VAL("-2.88"), X=VAL("2.88E-10")

関数 STR\$(式)

機能 式の数値を文字列に直す

例 A\$=STR\$(8), A\$=STR\$(2.88E-10)

関数 SPC(式)

機能 長さ<式>個の空白を与える

例 PRINT SPC(10); "FUKUYAMA"

関数 SPACE\$(式)

機能 長さ<式>の空白文字列を与える

例 A\$="福山": B\$="大学"  
PRINT A\$+SPACE\$(2)+B\$

関数 STRING\$(式, 文字)

機能 <式>個の文字の文字列を与える

例 PRINT STRING\$(70, "-")

関数 DATES

機能 1) 内蔵タイマーの日付を与える (YY/MM/DD)  
2) 内蔵タイマーの日付を変える 絶対に使用不可

例 1 PRINT DATES

関数 TIMES

機能 1) 内蔵タイマーの時間を与える (HH:MM:SS)  
2) 内蔵タイマーの時間を変える 絶対に使用不可

例 1 PRINT TIMES

## 例文

### REI10-1 文字列の結合と分解

A\$ = "FUKUYAMA-"

B\$ = "HEISEI-"

C\$ = "Univ."

D\$ = A\$ + B\$ + C\$

PRINT D\$

PRINT LEFT\$(D\$, 8)

PRINT MID\$(D\$, 10, 6)

PRINT RIGHT\$(D\$, 5)

### REI10-2 時刻から秒数の計算

X\$ = TIMES

PRINT X\$

H\$ = LEFT\$(X\$, 2)

S = 60 \* VAL(MID\$(X\$, 4, 2)) + VAL(RIGHT\$(X\$, 2))

PRINT H\$; "時から"; S; "秒経過しています。"

### REI10-3 ASCII コードと文字

A\$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

FOR I = 1 TO 26

B\$ = MID\$(A\$, I, 1)

PRINT USING "####"; ASC(B\$);

NEXT I

PRINT

```

FOR I = 1 TO 26
  B$ = MID$(A$, I, 1)
  PRINT USING "& &"; HEX$(ASC(B$));
NEXT I
PRINT
FOR I=&H41 TO &H5A
  PRINT CHR$(I);
NEXT I
PRINT

```

#### REI10-4 入力待ち

```

PRINT "文字キーを押して下さい。"
A$ = INKEY$
WHILE A$ = ""
  A$ = INKEY$
WEND
PRINT "押されたキーは"; A$; "です。"

```

#### REI10-5 文字列の削除

```

A$ = "This notebook is your notebook."
PRINT A$
B$ = "note": LB = LEN(B$)
N = INSTR(1, A$, B$)
WHILE N > 0
  A$ = LEFT$(A$, N - 1) + RIGHT$(A$, LEN(A$) - (N - 1) - LB)
  N = INSTR(N, A$, B$)
WEND
PRINT A$

```

#### REI10-5 の解説

文字列の中の note という語を削除して表示する。

利用する関数

LEFT\$(文字列,文字数)

RIGHT\$(文字列,文字数)

LEN(文字列) 文字列の長さを与える

INSTR(数値,文字列 1,文字列 2)

文字列 1 の数値番目の文字から検索を始めて、先頭から何文字目に文字列 2 があるか知らせる。文字列 2 を含まない場合は 0

方法

- 1) 削除対象の文字列を B\$とする
- 2) 最初に LB = LEN(B\$)で B\$の長さを求めておく
- 3) A\$ の先頭から何文字目に B\$ があるか見つける
- 4) 文字列のありかを見つけたら、  
A\$ = LEFT\$(A\$, N-1) + RIGHT\$(A\$, LEN(A\$) - (N-1) - LB)  
と置き換え、N の位置から再度検索する。
- 5) 4)から繰り返し、N = 0 になったらやめる。

## 演習 1 0

- 1) プログラム実行時の時間を 00 時 00 分 00 秒 と表示するプログラムを作れ。
- 2) プログラム実行時が、午前であるか午後であるか判定するプログラムを作れ。
- 3) 文字列"This is miss Lisa's island."の中に、何個"is"があるか数えるプログラムを作れ。
- 4) 文字列中の小英文字を大英文字に全て変えて表示するプログラムを作れ。  
例) ABadcgBCghi -> ABADCGBCGHI  
ヒント: 小文字の ASCII コードから 32 を引くと大文字の ASCII コードになる。
- 5) ESC キー(ASCII コード:27)が押されるまで、押されたキーの ASCII コードを表示するプログラムを作れ。 使用する関数: INKEY\$, ASC( )
- 6) REI10-5 を何個"note"が消されたかも表示するように改良せよ。
- 7) キーボードから入力された任意の文字列中の、ある単語を他の単語に置換するプログラムを作れ。  
例) That boy is my boyfriend. -> That girl is my girlfriend.

# ASCIIコード表

## 上位4ビット

	0	1	2	3	4	5	6	7	A	B	C	D
0	NUL	DLE	SP	0	@	P	`	p		-	タ	ミ
1	SOH	DC1	!	1	A	Q	a	q	。	ア	チ	ム
2	STX	DC2	"	2	B	R	b	r	「	イ	ツ	メ
3	ETX	DC3	#	3	C	S	c	s	」	ウ	テ	モ
4	EOT	DC4	\$	4	D	T	d	t	、	エ	ト	ヤ
5	ENQ	NAC	%	5	E	U	e	u	・	オ	ナ	ユ
6	ACK	SYN	&	6	F	V	f	v	ヲ	カ	ニ	ヨ
7	BEL	ETB	'	7	G	W	g	w	ア	キ	ヌ	ラ
8	BS	CAN	(	8	H	X	h	x	イ	ク	ネ	リ
9	HT	EM	)	9	I	Y	i	y	ウ	ケ	ノ	ル
A	LF/NL	SUB	*	:	J	Z	j	z	エ	コ	ハ	レ
B	VT	ESC	+	;	K	[	k	{	オ	サ	ヒ	ロ
C	FF	FS	,	<	L	¥	l		ヤ	シ	フ	ワ
D	CR	GS	-	=	M	]	m	}	ユ	ス	ハ	ン
E	SO	RS	.	>	N	^	n	~	ヨ	セ	ホ	・
F	SI	US	/	?	O	_	o	DEL	ツ	ソ	マ	°

## 下位4ビット



## 1 1 章 サブルーチン

### サブルーチン構文

構文 GOSUB ラベル

.....  
ラベル:

.....  
RETURN

機能 プログラム内で特定の処理をまとめて定義しておく

例 GOSUB LABEL

備考 まとまった処理ごとにサブルーチンにまとめておく  
サブルーチンからサブルーチンを呼ぶことも可  
メインルーチンとサブルーチンの変数のダブりに注意

### 例文

REI11-1 階乗を求めるサブルーチンと組合わせ  $nCr$  の計算 ( $nCr = n! / r!(n-r)!$ )

```
INPUT "n = ",N
INPUT "r = ",R
SBX=N: GOSUB FACT: A=SBK
SBX=R: GOSUB FACT: B=SBK
SBX=N-R: GOSUB FACT: C=SBK
P=A/B/C
PRINT "nCr =" ;P
END
'以下はサブルーチン
FACT: 'X! を計算し K で返す。
SBK=1
IF SBX<=1 THEN RETURN
FOR S=2 TO SBX
    SBK=SBK*S
NEXT S
RETURN
```

## 演習 1 1

- 1) サブルーチンコールすると、“Quick BASIC” と表示するサブルーチンを作成し、動作を確認せよ。
- 2) 変数 SBR に半径の値を与えてサブルーチンコールすると、円周と円の面積を出力するサブルーチンを作成し、動作を確認せよ。
- 3) 変数 SBM に行数、SBN に列数、SBA\$ に文字列を与えてサブルーチンコールすると、文字列 SBA\$ を SBM 行 SBN 列の行列のように表示するサブルーチンを作成し、動作を確認せよ。
- 4) 変数 SBX, SBN にそれぞれ実数と正の整数のデータを与えてサブルーチンコールすると、SBANS に以下の式の値を返すサブルーチンを作成し、動作を確認せよ。

$$SBANS=1 + SBX^1 + SBX^2 + SBX^3 + \cdots +SBX^{QBN}$$

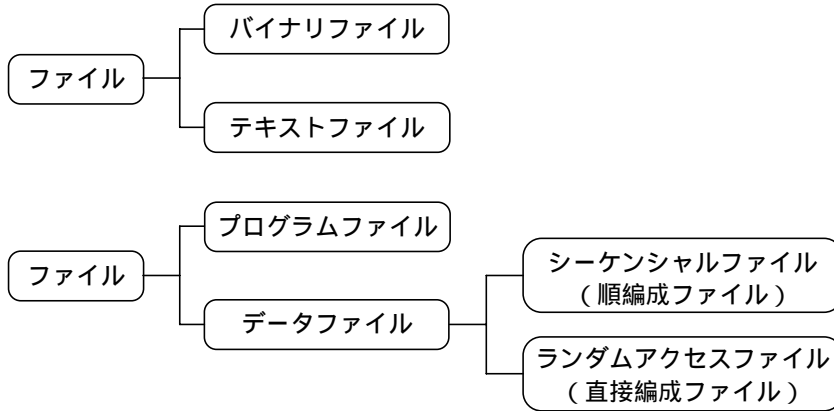
- 5) 演習 1 0 で作成した解答をサブルーチン化して

問題番号 >>

と聞いて番号を入力したら、その解答が表示され、再度問題番号の入力に戻るよ  
うにせよ。ただし、問題番号の入力で 0 を入力すると、プログラムが終了するも  
のとする。

## 12章 ファイル処理

### ファイルの分類



### 順編成ファイル

1. ファイルのオープン  
読み込み / 書き込み / 追加 のためのオープン (INPUT OUTPUT APPEND)
2. 読み込み / 書き込み / 追加
3. ファイルのクローズ

### ファイル入出力コマンド

命令 OPEN ファイル名 [OPEN モード] AS #ファイル番号  
機能 ファイルのオープン  
例 OPEN "TEST.DAT" FOR INPUT AS #1  
備考 OPEN モード: INPUT, OUTPUT, APPEND ファイルへの入出力の始めに必ず実行

命令 CLOSE #ファイル番号  
機能 ファイルのクローズ  
例 CLOSE #1  
備考 ファイルへの入出力の終りに必ず実行

命令 PRINT #ファイル番号 [, 出力並び]  
PRINT #ファイル番号, USING フォーマット文字列; 出力並び  
機能 指定ファイルに出力  
例 PRINT #1, A\$; B  
PRINT #1, USING "& & ####"; A\$; B

命令 INPUT #ファイル番号, 変数名 [, 変数名, ...]  
機能 指定ファイルから入力

例 INPUT #1,A

命令 LINE INPUT #ファイル番号,文字列変数名

機能 指定ファイルから 1 行まるごと入力

例 LINE INPUT #1,A\$

関数 EOF(ファイル番号)

機能 ファイル番号のファイルの終りで真を返す。

例 WHILE NOT EOF(1)

## 例文

REI12-1 ファイルへの文字列の書き込みとファイルからの読み込み

```
OPEN "A:¥DATA1.DAT" FOR OUTPUT AS #1
```

```
PRINT #1, "岡山,倉敷,福山,三原,広島"
```

```
CLOSE #1
```

```
OPEN "A:¥DATA1.DAT" FOR INPUT AS #1
```

```
FOR I = 1 TO 5
```

```
    INPUT #1, A$
```

```
    PRINT A$; " ";
```

```
NEXT I
```

```
PRINT
```

```
CLOSE #1
```

```
OPEN "A:¥DATA1.DAT" FOR INPUT AS #1
```

```
LINE INPUT #1, A$
```

```
PRINT A$
```

```
CLOSE #1
```

REI12-2 ファイルへの数字の書き込みとファイルからの読み込み

```
OPEN "A:¥DATA2.DAT" FOR OUTPUT AS #1
```

```
FOR I=1 TO 10
```

```
    PRINT #1, I;
```

```
NEXT I
```

```
CLOSE #1
```

```

OPEN "A:¥DATA2.DAT" FOR INPUT AS #1
FOR I = 1 TO 10
  INPUT #1, A
  PRINT A;
NEXT I
PRINT
CLOSE #1

```

```

OPEN "A:¥DATA2.DAT" FOR INPUT AS #2
WHILE NOT EOF(2)
  INPUT #2, A
  PRINT A;
WEND
PRINT
CLOSE #2

```

## 読み込み可能なデータ

### 数値

123,2, 7 4 , 2  
 区切り記号： , RETURN 1つだけ  
 SPACE いくつでも( , RETURN と組合せてもよい)  
 ,, のような NUL の処理 PC9800: 0 を代入 FMR: エラー

### 文字列

ADF,ADFA ,ADFADFF , FF  
 区切り記号： , RETURN 1つだけ(SPACE は文字列の一部と解釈される)  
 ,, のような NUL の処理 "" (ヌル文字列) を代入  
 1行丸ごと読み込む場合は LINE INPUT 命令参照

## 演習 1 2

- 1) データファイル "A:¥DEX1.DAT" に "Quick BASIC" と出力するプログラムを作成せよ。
- 2) データファイル "A:¥DEX2.DAT" に 1~100 の整数乱数を 4桁右詰で出力するプログラムを作成せよ。
- 3) データファイル "A:¥DEX1.DAT" から文字を読み込み、表示するプログラムを作成せよ。

- 4) データファイル "A:¥DEX2.DAT" から数値を 100 個、数を指定して読み込み、表示するプログラムを作成せよ。
- 5) データファイル "A:¥DEX2.DAT" から、ファイルの終わりを確認しながら数値を読み込み、その平均値を表示するプログラムを作成せよ。
- 6) キーボードからファイル名を入力すると、そのテキストファイルの内容を表示するプログラムを作成せよ。
- 7) 売上比較のデータファイル "A:¥DEX3.DAT" に以下の文字列と数値が保存されているとする。

```
5 ..... データ数
相原,4000
竹熊,3200
吉田,4800
中崎,2800
なにわ,2000
```

このデータをファイルから読み込んで、棒グラフを書くプログラムを作成せよ。グラフのデザインは各自に任す。ただし、数値の単位は万円である。プログラムはサブルーチンやコメント等を用いて出来るだけ読み易くすること。

**【参考図書】**

1) 最新はじめての Quick BASIC

河西朝雄著 技術評論社

2) ナツメ/ハンディリファレンス 37・38

Quick BASIC ハンドブック vol.1 基本操作 vol.2 リファレンス

阿部友計著 ナツメ社