

課題 3

次の各プログラムを，各行の意味を十分理解しながら実行して，結果をレポートにまとめよ．ここで用いられる命令は最も基本的なものだから，これらを十分理解したうえで次の課題に進むこと．

プログラム 1 (四則演算)

```
10 rem rem input print + - * /, end =
20 rem four rule
30 input a,b
40 print a,b
50 c=a+b
60 d=a-b
70 e=a*b
80 f=a/b
90 print c,d,e,f
100 end
```

各行の命令の意味は

10, 20 注釈文．プログラムには影響しない
30 変数 a,b を入力せよという命令
40 変数 a,b の内容を画面上に表示せよという命令
50 a と b の和を c に代入せよという命令
60 a と b の差を d に代入せよという命令
70 a と b の積を e に代入せよという命令
80 a と b の商を f に代入せよという命令

プログラム 2 (データ読み込み)

次のプログラムでは a\$, b\$, c\$, d\$ は数値ではなくて文字列を読み込む変数である。数値を読み込む変数 a,b,c,d とのプログラムないでの取り扱いの違いを確認せよ。

```
10 rem read data restor
20 rem character data
30 read a,b,c,d
40 print a,b,c,d
50 print a+b,c-d
60 print
70 restore
80 read a,b,c,d
90 print a,b,c,d
100 print a+b,c+d
110 end
120 data 1,2.345,-1,-2.345
```

各行の意味は

30 120 行の data 以下の各数値を変数 a,b,c,d に読み込めという命令
70 再度 data 以下を読み込む準備をせよという命令

プログラム 3 (一つの繰り返し)

N に様々な値をいれて実行してみよ.

```
10 rem for to step next input " " ; ^
20 rem for next
30 input "input N";N
40 print N
50 s=0
60 for i=1 to N step 2
70 s=s+i^2
80 print i,s
90 next i
100 end
```

幾つか解説すると

input "input N";N と書くと実行したとき ? ではなく input N? と出力される. 何を入れるのかが解るように書きたいとき使用すればよい. 書きたい文字列は” ”で囲む.

50 s=0 はいったん s に 0 を代入する.

60 for i=1 to N step 2 は 90 next i と組み合わせて使う. 変数 i が 1 から N まで 2 増で動く. step 3 とすると 3 増となる. step を省略すると 1 増しになる.

70 i^2 は i の二乗という意味. i^3 と書くと i の三乗です. $s=s+i^2$ は s に i^2 を加えたものを s と置き直していること.

従って 10 行から 90 行で i が N までの奇数を動くときの 2 乗の和を求めていることになる.

プログラム 4 (三つの繰り返し)

次のプログラムが何をするプログラムであるかもレポートにまとめよ.

```
10 rem triple for next
20 rem binary-decimal transformation
30 print "binary","decimal"
40 for i = 0 to 1
50 for j = 0 to 1
60 for k = 0 to 1
70 l = 22*i+2*j+k
80 print i;j;k,l
90 next k
100 next j
110 next i
120 end
```

プログラム 5 (条件判断繰り返し)

```
0 REM let while wend print ; < > do loop until
10 REM Test WHILE-WEND LOOP
20 PRINT "START"
30 LET X = 0
40 WHILE X < 7
50 PRINT "x is ";X
60 LET X = X + 1
120 WEND
130 PRINT "WHILE-WEND LOOP END"
140 REM Test DO LOOP-UNTIL
150 PRINT "START"
160 I=0
170 DO
180 I=I+2
190 PRINT "i is";I
200 LOOP UNTIL I>7
210 PRINT "DO LOOP-UNTIL END"
```

いままでは英小文字でしたが、英大文字で書いてもかまわない。但し Basic によっては大文字と小文字を区別するものもありますので注意が必要。

40 WHILE X < 7 と 70 行目の WEND を組として使用する。意味は X < 7 である限り WEND の一つ前の行までを繰り返す。

170 DO と 200 行目の LOOP UNTIL を組として使用する。意味は I>7 になるまで LOOP UNTIL のある行の一つ前の行までを繰り返す。条件が最後にあるので少なくとも一回は実行される。

プログラム 6 (1次元配列)

このプログラムは10個の入力データのヒストグラムを書く。

```
10 rem dim ( ) tab( )
20 rem histogram
30 dim d(10)
40 for i=1 to 10
50 input d(i)
60 next i
70 for i=1 to 10
80 n=d(i)
90 s$=""
100 for j=1 to n
110 s$=s$+"*"
120 next j
130 print tab(2);i;tab(6);n;
140 print tab(12);s$
150 next i
160 end
```

30 dim d(10) で d(1),d(2),...,d(10) という変数ができるように宣言する。

90 s\$="" は s\$ に何も無い文字列 (空) を代入している。

110 s\$=s\$+"*" は s\$ に文字列として*を加えたものを s\$ と置き直している。

130 tab(2) は空白を2つとるとのこと。

80 から 110 行では s\$ は*が d(i) 個の*****の様な形になっていることがわかる。

プログラム 7 (二次元配列)

次のプログラムは n 人 (最大 10 人) の人が 3 科目受験したときの個人の平均と科目の平均について出力する.

```
10 rem dim ( , )
20 rem array
30 dim K(10,3)
40 input n
50 for i = 1 to n
60 input K(i,1),K(i,2),K(i,3)
70 next i
80 print "no","heikin";
90 print "1","2","3"
100 for i=1 to n
110 l = 0
120 for j=1 to 3
130 l = l+K(i,j)
140 next j
150 h = l/3
160 print i,h,K(i,1),K(i,2),K(i,3)
170 next i 180 print "kamoku";
190 print "heikin"
200 for j = 1 to 3
210 l = 0
220 for i = 1 to n
230 l = l+K(i,j)
240 next i
250 h = l/n
260 print j,h
270 next j
280 end
```

30 dim K(10,3) は $K(i,j)$ ($i=1,2,3,\dots,10,j=1,2,3$) である 30 個の変数を使えるように宣言している.

50 行から 70 $K(i,1)$ が科目 1 $K(i,2)$ が科目 2, $K(i,3)$ が科目 3 で $i=1,2,3,\dots,10$ までデータを入れる. 100 から 170 で個人の平均と点数を出す. 200 から 270 で科目の平均を出す.

プログラム 8 (分岐文)

```
10 rem goto if then else sqr
20 rem if statement
30 input a,b,c
40 if a<=0 then goto 130
50 print
60 print "a,b,c=";a,b,c
70 if a+b<c then goto 80
71 if b+c<a then goto 80
72 if c+a<b then goto 80 else goto 90 80 print "impossible"
85 goto 30
90 t=.5*(a+b+c)
100 s=sqr(t*(t-a)*(t-b)*(t-c))
110 print s
120 goto 30
130 end
```

40 if a<=0 then goto 130 は”もし a が 0 以下であるときは then 以下を実行せよ, そうでなければ次の行に行け” という意味. goto 130 とは”130 行目に行け”ということ. 従ってこの 40 行目は”もし a が 0 以下であるときは 130 行目に行け”ということになり, プログラムは終了する.

a>=0 は a は 0 以上を表す.

70 から 72 は上記の事から a,b,c が三角形になるための条件をチェックしていることになる.

85 goto 30 三角形が作られないときはもう一度 a,b,c の入力に戻る.

90 t=.5*(a+b+c) の .5 は 0.5 の事.

100 s=sqr(t*(t-a)*(t-b)*(t-c)) の sqr は () の中の平方根の意味. 従って 90 から 100 ではヘロンの公式で三角形の面積を求めていることになる.

プログラム 9 (サブルーティン)

所謂サブルーティンの演習である。サブルーチンはあるプログラムの中で何度も使われるなどの一般性のある処理部分をまとめて作っておき、必要に応じて呼び出して用いる。次のプログラムは2次方程式の三つの係数 a, b, c を入れて、判別式を計算して実根を持つ場合にその根を書き、複素根を持つときは complex root と出力する。

```
10 rem gosub return -1*
20 rem subroutine
30 input a,b,c
40 if a=0 then goto 220
50 print a,b,c
60 gosub 180
70 if d<0 then goto 80 else goto 100
80 print "complex root"
90 goto 30
100 y=sqr(d)/a2
110 z=x+y
120 print z
130 z=x-y
140 print z
150 print
170 goto 30
180 a2=2*a
190 x=-1*b/a2
200 d=b^2-4*a*c
210 return
220 end
```

60 gosub 180 で 180 行目に行き、180-200 (この部分がサブルーティン) を実行する。そして、次の行の return に出会うと、60 行目 (gosub のある行) の次の 70 行目の間に戻る。gosub と return はこの様に組にして用います。

190 $x=-1*b/a2$ は $b/a2$ の符号を変えたもの。他の basic program では $-b/a2$ と書くこともある。我々の basic では $-b/a2$ と書くとエラーが生じますので注意すること (実際に書いて確かめよ)。

プログラム 10 (場合分け)

```
0 rem on n goto on n gosub return
10 rem Test ON...GOTO Statement
20 input "Enter a number 1-3:";n
30 on n goto 40, 60, 80
40 print "end program"
50 goto 150
60 print "next test"
70 goto 100
80 print "once more test"
90 goto 20
100 print "ON...GOTO end"
110 rem Test ON..GOSUB Statement
120 input "Enter a number 1-2";n
130 on n gosub 160, 180
140 print "ON..GOSUB END"
150 end
160 print "You entered 1"
170 return
    180 print "You entered 2"
190 return
```

20 input "Enter a number 1-3:";n で n に 1,2,3 のいずれかを入れる。

30 on n goto 40, 60, 80 ではさっき入れたその番号により行き先が変わる。具体的には1のとき40行目,2のとき60行目,3のとき80行目に行き、その行から実行する。on と goto を組にして用いる。

120 input "Enter a number 1-2";n で n に 1,2 のいずれかを入れる。

130 on n gosub 160, 180 ではさっき入れたその番号により行き先が変わる。具体的には1のとき160行目,2のとき180行目に行き、その行から実行する。最後に return に出会ったら on gosub のある次の行に戻る。on gosub と return を組にして用いる。

プログラム 11 (関数)

組み込み関数と関数の定義のプログラムの例.

```
60 REM abs atn cos log sin tan sgn int exp rnd def fn
70 REM
80 REM VALUE OF SOME FUNCTION
90 INPUT "POSITIVE NUMBER X";X
100 PRINT "ABS(-X): "; ABS(-1*X)
110 PRINT "ATN(-X): "; ATN(-1*X)
120 PRINT "COS(-X): "; COS(-1*X)
130 PRINT "LOG(X): "; LOG(X)
140 PRINT "SIN(-X): "; SIN(-1*X)
150 PRINT "SQR(X): "; SQR(X)
160 PRINT "TAN(-X): "; TAN(-1*X)
170 PRINT "SGN(-X): "; SGN(-1*X)
180 PRINT "INT(-X): "; INT(-1*X)
190 PRINT "EXP(-X): "; EXP(-1*X)
200 PRINT "RND(-X): "; RND(-1*X)
210 REM DEFINE FUNCTION
220 INPUT "A,B";A,B
220 DEF FNADD( X, Y ) = X+Y
230 DEF FNHYPS( X ) = (EXP(X)-EXP(-1*X))/2
240 PRINT "A+B=";FNADD( A, B)
250 PRINT "HYPSIN(A)=";FNHYPS( A )
500 END
```

よく用いられる組み込みの関数を以下に書いておく.

- 100 ABS(-1*X) は-1*Xの絶対値,
- 110 ATN(-1*X) は-1*Xの tan の逆関数,
- 120 COS(-1*X) は-1*Xの cos,
- 130 LOG(X) は Xの log,
- 140 SIN(-1*X) は -1*Xの sin,
- 150 SQR(X) は Xの平方根,

160 TAN(-1*X) は-1*X の tan,

170 SGN(-1*X) は-1*X が正であれば 1, 負であれば-1,0 であれば 0,

180 INT(-1*X) は-1*X の整数部分,

190 EXP(-1*X) は-1*X の指数関数,

200 RND(-1*X) は -1*X の値の乱数の値を表す.

220-250 は手作りの関数の作り方を示す.

220:DEF FNADD(X, Y) = X+Y で FNADD(X, Y) という関数を X+Y で定義している.

230:DEF FNHYPS(X) = (EXP(X)-EXP(-1*X))/2 で FNHYPS(X) という関数を右辺で定義している. 従って hyperbolic sin を定義していることになる.

手作りの関数を定義するときには DEF FN... の様に関数の名前の最初は FN とするよう to すること.

プログラム 12 (仕上げ)

最後の仕上げとして、N までの素数をプリントするプログラムである。どうして素数が生成されるか理解したうえで実行すること。

```
10 rem mod
20 rem prime number
30 DIM A(200)
40 print
50 N=200
60 for i=2 to N
70 A(i)=1
80 next i
90 for i=2 to SQR(N)
100 for j=i+1 to N
110 if (j mod i)=0 then A(j)=0
120 next j
130 next i
150 for i=2 to N
160 if A(i)=1 then print i;
170 next i
180 print
190 end
```

60-80 で $A(i)$ に 1 を代入する。

110 if (j mod i)=0 then A(j)=0 に現れている $j \bmod i$ は j を i で割った余りを表す。従って 110 は”もし j が i で割り切れるならば $A(j)$ に 0 を代入せよ” という命令になる。

以上から $A(j)$ が 2 以上の数で割れたら $A(j)=0$ とするわけだから、これを繰り返せば j が素数のときは $A(j)=1$, そうでなければ $A(j)=0$ となる。

150-190 で $A(i)=1$ のときの i をプリントする。