

プログラムのページ

担当 森 口 繁 一

本号から本誌に「プログラムのページ」を設けることになりました。目的は、プログラム——特に科学技術計算・経営数学計算などの各種計算方式——について、会員相互の間で情報を交換し、相携えてこの分野の発展に貢献しようという所にあります。以下に述べる投稿規定および例をご参照の上、ふるってご投稿下さい。

投稿規定

(1) 何かの機械で実際に通したことのあるプログラムに限る（その際に用いた言語は ALGOL でなくてもよい）。

(2) 記述は ALGOL 60 の文法に従うこと。[ALGOL 60 については、本誌第1巻、第3号(1960) 157～162 ページ、第1巻、第4号(1960) 219～227 ページ、および第2巻、第1号(1961) 30～38 ページを参照。]

(3) 入出力については、さしあたり、以下の例に見られるように、次の四つの基本手続きを標準として認めることとする。

READ (X) 紙テープ（またはカード）から1個の数値を読み、その値を変数Xに与える。

PRINT (X) Xの値を標準の様式で印刷する。

PRINTSTRING ('……') ……のところにある文字や数字のつながり(string)をそのまま印刷する。

CRLF 復帰改行 (carriage return, line feed) を行なう。

(4) 始めに問題および解法の要旨を日本語で説明し、その次にプログラムを記述し、必要ならばそのあとに注をつけること。

例 今回は第1回目なので、ていさいの見本という意味と、ALGOL 60 でのやさしいプログラム例という意味と、両方を兼ねていくつかの例をならべてみる。

6201. 四則

森口繁一（東大工学部）

データ A, B から読み取り、和 S, 差 D, 積 P, および商 Q を求め、これらを印刷する。

```
begin real A, B, S, D, P, Q;  
  READ(A); READ(B);  
  S:=A+B; D:=A-B;  
  P:=A×B; Q:=A/B;  
  CRLF; PRINT(A); PRINT(B);  
  CRLF; PRINT(S); PRINT(D);
```

PRINT(P); PRINT(Q)

end

[注 1] ここでは A, B, S, D, P, Q はすべて実数(浮動小数点表示の数)とした。もしこれらを整数として扱いたいならば、上のプログラムで real のところを integer に変え、 $Q:=A/B$ を $Q:=A \div B$ に改めればよい。

[注 2] ALGOL 60 で書かれたプログラムはすべて begin で始まり end で終る。そして begin に続いて宣言(declaration)が行われる。プログラム中に現われる変数はすべて、始めにその型——実数か整数か——を宣言しておかねばならない。:= は、右辺を計算し、その値を左辺の変数に与えよという意味である。宣言や実行命令を区切る区切符号は ; (セミicolon) である。ただし begin の直後と end の直前には ; は要らない。

6202. 最小値

森口繁一（東大工学部）

データ $a_1, a_2, \dots, a_n$ を紙テープ上から読み取り、これらを印刷し、かつ、そのうちの最小値とデータの個数とを印刷する。ここにデータは、1,000 を越えない正の整数であり、データの尽きた印としては 0 が穿孔してあるものとする。

```
begin integer A, MIN, N;
```

```
  MIN:=1,000; N:=0;
```

```
  L1: READ(A);
```

```
    if A=0 then go to L2;
```

```
    CRLF; PRINT(A);
```

```
    if A<MIN then MIN:=A;
```

```
    N:=N+1; go to L1;
```

```
  L2: CRLF; CRLF; PRINT(MIN); PRINT(N)
```

```
end
```

[注 1] L1, L2 は立て札 (label) として使っている。この例では、go to L1 はループを反復するためにともどることにあたり、go to L2 はループから脱出することにあたる。

〔注 2〕 if~then……は、「もし~ならば…」という意味であり、~が真のときだけ……が実行される。この……は一般に；で終る（いまの例で $N:=N+1$ は、 $A<MIN$ という条件が満たされても満たされなくても実行される）。

6203. 積の和

森口繁一（東大工学部）

データ $a_1, a_2, \dots, a_{20}, b_1, b_2, \dots, b_{20}$ が、この順に紙テープ上に穿孔してあるとき、これを読み込み、それから a_1, b_1 を 1 行に、 a_2, b_2 を次の 1 行に、……というふうに印刷し、かつ積の和 $s=a_1b_1+\dots+a_{20}b_{20}$ を計算して印刷する。

```
begin integer I; real S;
  array A, B [1:20];
  for I:=1 step 1 until 20 do READ(A[I]);
  for I:=1 step 1 until 20 do READ(B[I]);
  S:=0;
  for I:=1 step 1 until 20 do
    begin CRLF; PRINT(A[I]); PRINT(B[I]);
    S:=S+A[I]*B[I] end;
  CRLF; CRLF; PRINT(S)
end
```

〔注 1〕 for I:=1 step 1 until 20 do ……は、「I を 1 から 1 きざみで 20 まで変え、その各 I に対して……を実行せよ」ということである。この……も；で終る。ただし……が begin ~; ~; ~ end; の形をしているときは、途中の；では終わらないで end のあとの；で終ることになる。

〔注 2〕 単に array と宣言された場合は、real array（実数値をとる配列）と宣言されたのと同様に解される。この例で、array A, B [1:20] は A [I] と B [I] が、共に I=1 から I=20 までの範囲に添字の動くベクトルであることを宣言している。

6204. 多項式

森口繁一（東大工学部）

多項式 $p(x)=a_0+a_1x+\dots+a_nx^n$ の、次数 n および係数 $a_0, a_1, \dots, a_n$ は紙テープ上に穿孔されているものとして、これを読み取り、 x を 0 から 0.02 きざみで 0.50 まで変えて、各 x に対して $p(x)$ を計算し、 x と $p(x)$ とをならべて表を作る。

```
begin integer N, I; real X, P;
  array A[0:N];
  PREP: READ(N);
  for I:=0 step 1 until N do
```

```
  READ(A[I]);
```

```
  for X:=0 step 0.02 until 0.50 do
```

```
    begin P:=A[N];
```

```
      for I:=N-1 step -1 until 0
```

```
        do P:=P*X+A[I];
```

```
      CRLF; PRINT(X); PRINT(P)
```

```
    end end
```

〔注 1〕 立て札 PREP: は、準備(preparation)という気持ちで、いわば説明用につけたに過ぎない。

〔注 2〕 最後に end が二つあるが、その一つは下から 5 行目の begin に対するもの、他の一つは一番上の begin に対するものである。

〔注 3〕 2 進法の機械では、換算の際の丸め誤差のせいで、どうかすると最後の $X=0.50$ に対する実行が実現しないで終ることがある。これを防ぐには、この例では……until 0.51 do……のように、ちょっとだけ大きい数を書いておけばよい。（制御される変数が整数の場合はこういう心配はない。）

6205. 逆正弦函数表

森口繁一（東大工学部）

$0 \leq x < 1$ のとき近似式 $\arcsin x = \pi/2 - \sqrt{1-x} P(x)$ を用い、 $x < 0$ については $\arcsin x = -\arcsin |x|$ として計算する。ここに $P(x)$ は x の多項式であり、その次数 n と係数 $a_0, a_1, \dots, a_n$ は紙テープ上から読み込むものとする。 x は -0.95 から 0.05 きざみに 0.95 まで変え、 x と $\arcsin x$ とをならべた表を作る。

```
begin integer N, I; array A[0:N];
```

```
PREP: READ(N);
```

```
  for I:=0 step 1 until N do
```

```
    READ(A[I]);
```

```
  begin real procedure P(X);
```

```
    value X; real X;
```

```
    begin real PP; PP:=A[N];
```

```
      for I:=N-1 step -1 until 0
```

```
        do PP:=PP*X+A[I]; P:=PP
```

```
    end;
```

```
    real X, ABSX, ACSINX, HFPI;
```

```
    HFPI:=1.570796327;
```

```
    for X:=-0.95 step 0.05 until 0.95 do
```

```
      begin ABSX:=ABS(X);
```

```
        ACSINX:=
```

```
          HFPI-P(ABSX)*SQRT(1.0-ABSX);
```

```
          if X<0 then ACSINX:=-ACSINX;
```

```
CRLF; PRINT(X); PRINT(ACSINX)
end end end
```

〔注 1〕 $P(x)$ を計算する部分を **procedure** にしてみた。これは閉じたサブルーチンとして編集される。real **procedure** として $P(X)$ を宣言したので、これが関数記号として式中に使えることになる。

〔注 2〕 HFPI は half pi という気持で、 $\pi/2$ を表わす。

〔注 3〕 関数 $ABS(X)$ および $SQRT(X)$ は、それぞれ X の絶対値 (absolute value) および平方根 (square root) を与えるもので、標準函数中に含まれているので、宣言しないで使ってよい。なお、 $ABSX := ABS(X)$ は、 X の絶対値を $ABSX$ という一つの変数記号で表わすという意味である。

6206. 2次方程式

森口繁一 (東大工学部)

2次方程式 $ax^2+2bx+c=0$ の係数 a, b, c を n 組、テープに穿孔して与える (その頭には n を穿孔しておく)。これを読み込んで根を計算し、 a, b, c を印刷したあとに、実根の場合はそのまま、複素根の場合は実部と虚部を、間に $+$ をはさみ、虚部のあとに I をつけて印刷する。

```
begin real A, B, C, D, S, T, X1, X2;
      integer I, N;
READ(N);
for I:=1 step 1 until N do
```

```
begin READ(A); READ(B); READ(C);
      CRLF; PRINT(A); PRINT(B); PRINT(C);
      D:=B↑2-A×C;
      if D≥0 then begin S:=-B/A;
                    T:=SQRT(D)/A; X1:=S+T;
                    X2:=S-T; PRINT(X1);
                    PRINTSTRING(' ', '#'); PRINT(X2)
      end else begin X1:=-B/A;
                    X2:=SQRT(-D)/A; PRINT(X1);
                    PRINTSTRING('+-'); PRINT(X2);
                    PRINTSTRING('I') end
end end
```

〔注 1〕 判別式 $D=b^2-ac$ が ≥ 0 のときは実根であるから、**then** に続く **begin** から **end** までを実行し、 $D < 0$ のときは複素根なので、**else** に続く **begin** から **end** までを実行する。string 中の # は間隔 (space) を示す。

〔注 2〕 $X1, X2$ は二つの変数記号であって、互に何の関係もない。もしこれの代わりに $X[1], X[2]$ と書くとするれば、それは一つの配列 (array) X の二つの要素ということになる。一般に x_1, x_2 などを ALGOL で表わすには、この二つのやり方のうちのどちらかが使われる。 x_1, x_2 などに関する演算が x_i に関する演算として一般の形に書けて、それを **for** $I:=\dots$ という形で制御しようというときは $X[I]$ という形がよく、その他のときは $X1, X2$ のような形の方がよい。