FORTRAN での Format-Free な入出力*

清水留三郎**　浦部雍子***

はじめに

FORTRANにおいては，カードから読み込んだり，用紙に印刷したりする入出力文（statement）には，書式（format）がかならず要ることになっている．書式は，できるかぎり多様なデータの入出力を可能にするためのものであるが，データの値さえ正しく受け渡しされれば，形がきまっていようとかまわない者にとっては，はなはだ厄介なものである．とくに入力においては，引続くデータの区切れさえわかれば，その値は明らかであるにもかかわらず，書式を指定しなければならないのは，特殊な場合のために標準の場合が犠牲にされている嫌いがある．現にFORTRANの講習会で，わかりにくく，誤り易いのが書式である．

そこでFORTRANの入出力用サブルーチンを作成して，区切りのついているデータならば，書式なしで読み込め，また任意のデータを標準書式で出力できるようにした．その際，出力サブルーチンで出力したデータは，入力サブルーチンで読み込めるように標準書式を定めた．

入出力サブルーチンの作成には，FORTRANの文字処理の機能を利用した．以下に入力と出力に分けて，その手法などを述べる．

1. 入　　力

1.1 データの表現形式

データの外部表現はFORTRANプログラム中の定数の表現形式と同じとした．ただし，底と指数との間の2個以内の空白を除いて+，間に空白をはさんではならない．引続くデータの間は，数字 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，符号 +, −，底 D, E，小数点．以外の文字で区切られるものとした．実際には空白またはコンマを区切りに使うのが標準であろう．データはカードの第1欄から第72欄までに穿孔するものとし，前のカードの第72欄と次のカードの第1欄との間には，空白が少なくとも1個あるものとした．すなわち，一つのデータは2枚のカードにまたがらないものとし，第72欄で終るデータは，区切りがなくても，自動的に次のデータと区切られる．

データに区切りを付けることによって，データの間隔を適当につめることができるので，とくに整数についてはせん孔が簡単になる．

1.2 入力サブルーチン

入力サブルーチンは次の5種類から成る：

```
ININTE(CHANEL, DESTIN)
DININT(CHANEL, DESTIN)
INREAL(CHANEL, DESTIN)
DINREA(CHANEL, DESTIN)
INSYMB(CHANEL, DESTIN)
```

ここで，CHANELはデータを取って来る装置の番号を示す整数で，DESTINはデータの読み込み先を示し，それぞれ整整型，倍長整数型，実数型，倍精度実数型および整数型の変数名である．入力サブルーチンの内容を**第1表**に示す．最後のINSYMBは，入力サブルーチンの基本となるもので，その他のサブルーチンのために，カード上の次の文字を対応する整数に変換したものを与えるが，これを直接利用することももちろんよい．最初の四つのサブルーチンはINSYMBを利用して外部表現を内部表現に変換する．

INSYMBは，初めて呼ばれたとき，カードを1枚読み込んで，第1欄から第72欄までの文字を，1文字ずつに分解して，1次元の配列INBUFに入れ，その最初の文字を対応する整数に変換してDESTINに与えてもどる．次に呼ばれたとき，INBUF中の次の文字を対応する整数に変換してDESTINに与えてもどる．これを第72欄の文字を変換するまで反復する．何欄目の文字を変換しているかは，共通ブロックFREEIN中の変数IBCCの値によってわかる．カード1枚分を読み終わった状態になって，その次に呼ば

---

* Format-Free Input/Output in FORTRAN, by Tomesaburo Simizu (Dept. of Pure and Applied Sciences, Coll. of Gen. Ed., Univ. of Tokyo) and Yasuko Urabe (Computation Center, Institute of JUSE)

** 東京大学教養学部基礎科学科
*** 日本科学技術研修所電子計算機センター

+ 標準書式で出力されたデータを入力できるようにするための例外である．

第 1 表

```
      SUBROUTINE INSYMB(CHANEL,DESTIN)
      INTEGER CHANEL,DESTIN
C     INPUT SYMBOL ( CHANNEL, DESTINATION
      COMMON/FREEIN/IBCC
      DIMENSION INBUF(72)
      INTEGER X
      IF(IBCC.NE.0) GO TO 10
C     READ THE NEXT CARD INTO INPUT BUFFER
      READ (CHANEL,100) INBUF
  100 FORMAT (72A1)
      IBCC=-72
C     CONVERT THE NEXT CHARACTER TO INTEGER
   10 IBCC=IBCC+1
      X=(INBUF(IBCC+72)-5H0     )/67108864
      IF(X.LT.0)X=X+64
      DESTIN=X
      RETURN
      END

      SUBROUTINE ININTE(CHANEL,DESTIN)
      INTEGER CHANEL,DESTIN
C     INPUT INTEGER ( CHANNEL, DESTINATION )
      DOUBLE LENGTH INTEGER X
      CALL DININT(CHANEL,X)
      DESTIN=X
      RETURN
      END

      SUBROUTINE DININT(CHANEL,DESTIN)
      INTEGER CHANEL
      DOUBLE LENGTH INTEGER DESTIN
C     INPUT DOUBLE LENGTH INTEGER ( CHANNEL, DESTINATION )
      COMMON/FREEIN/IBCC
      INTEGER CH,SW
      DOUBLE LENGTH INTEGER X
      LOGICAL S
      X=0
      SW=1
      S=.FALSE.
   20 IF(IBCC.EQ.0.AND.SW.EQ.2) GO TO 60
  100 CALL INSYMB(CHANEL,CH)
C     DIGIT
      IF(CH.LT.10) GO TO 10
C     BLANK
      IF(CH.EQ.10) GO TO (20,60),SW
C     MINUS
      IF(CH.EQ.32) GO TO (30,200),SW
C     D,E OR PERIOD
      IF(CH.EQ.21.OR.CH.EQ.20.OR.CH.EQ.27) GO TO 200
C     PLUS
      IF(CH.EQ.26)GO TO (40,200),SW
   60 IF (S) X=-X
      DESTIN=X
      RETURN
   10 X=X*10+CH
```

```
   40 SW=2
      GO TO 20
   30 S=.TRUE.
      GO TO 40
  200 WRITE(6,1000)
 1000 FORMAT (17H INPUT DATA ERROR)
      STOP
      END

      SUBROUTINE INREAL(CHANEL,DESTIN)
      INTEGER CHANEL
C     INPUT REAL NUMBER ( CHANNEL, DESTINATION )
      DOUBLE PRECISION X
      CALL DINREA(CHANEL,X)
      DESTIN=X
      RETURN
      END

      SUBROUTINE DINREA (CHANEL,DESTIN)
      INTEGER CHANEL
      DOUBLE PRECISION DESTIN
C     INPUT DOUBLE LENGTH REAL NUMBER ( CHANNEL, DESTINATION )
      COMMON/FREEIN/IBCC
      DOUBLE LENGTH INTEGER M
      INTEGER EX,PC,SW,CH
      LOGICAL MS,ES,P
      M=0
      EX=0
      MS=.FALSE.
      ES=.FALSE.
      P=.FALSE.
      SW=1
      PC=0
   50 IF(IBCC.EQ.0.AND.SW.NE.1) GO TO 1000
   20 CALL INSYMB(CHANEL,CH)
      IF(CH.GE.10) GO TO 30
C     C=9
      IF(SW.GE.3) GO TO 40
      M=M*10+CH
      PC=PC+1
   21 SW=2
      GO TO 50
   40 EX=EX*10+CH
   41 SW=5
      GO TO 50
C     BLANK
   30 IF(CH.EQ.10) GO TO (50,1000,60,60,1000),SW
C     .
      IF(CH.EQ.27) GO TO (70,70,1000,1000,1000),SW
C     -
      IF(CH.EQ.32) GO TO (80,81,81,82,1000),SW
C     PLUS
      IF(CH.EQ.26) GO TO (21,83,83,41,1000),SW
C     D,E
      IF(CH.EQ.20.OR.CH.EQ.21) GO TO (90,91,1000,1000,1000),SW
```

```
C     END MARK
 1000 IF(BS) EX=.EX
      IF(P) EX=EX-PC
      IF(MS) M=.M
      DESTIN=FLOAT(M)*10.0**EX
      RETURN
   60 SW=SW+1
      GO TO 50
   70 P=.TRUE.
      PC=0
      GO TO 21
   80 MS=.TRUE.
      GO TO 21
   81 BS=.TRUE.
   83 SW=4
      GO TO 50
   82 BS=.TRUE.
      GO TO 41
   90 M=1
   91 SW=5
      GO TO 50
      END
```

れると，次のカードについて前のカードと同じことを反復する．

カードの場合ばかりでなく，紙テープの場合でも，INSYMB というシステム・サブルーチンを用意しさえすれば，その他はまったくそのまま適用できる．

2. 出力サブルーチン

2.1 標準書式

標準書式は次の3種類とした:

整数型　I5, 3X

実数型　1P E12.5, 4X

文字型　5A

データの内部表現から外部表現への変換と印刷動作の指示は別個に独立に行なうものとし，印刷動作の指定のときに，改行を制御できるものとした．

2.2 出力サブルーチン

出力サブルーチンは次の4種類から成る:

```
OUTINT(LOGUNO, INTEGR)
OUTREA(LOGUNO, SOURCE)
OUTSTR(LOGUNO, STRING)
OUTLIN(LOGUNO, CARCON)
```

ここで，LOGUNO はデータを出力する先の装置の番号を示す整数で，INTEGR, SOURCE, STRING は出力すべき値を示す，それぞれ整数型，実数型および文字型の式，また CARCON は改行を制御する文字型の式である．

出力サブルーチンを**第2表**に示す．データの内部表現から外部表現への変換には，FORTRAN の変換機能を利用することにした．入力の場合のようにサブルーチンで変換を行なうよりも簡単だからである．

最初の三つのサブルーチンは，データの変換を指定するためのものであり，指定されたデータを共通ブロック OUTBUF 中の1次元の配列 DATA 中の次の位置に移動する，と同時に対応する標準書式を配列 FMT 中の次の位置に記録しておく．データを移動するときに，DATA 中に十分な余裕が残っていなければ，OUTLIN を呼んで前の記録を印刷して，DATA と FMT とを空にしてから同様にする．

最後の OUTLIN は，FORTRAN の書式付 WRITE 文を利用して，データ DATA を書式 FMT にしたがって出力する．

出力サブルーチンで問題になるのは，DATA を整数型か実数型のどちらかにきめなければならないので，たとえば実数型ときめると，整数型データを DATA に移動するときに，実数化が起こってしまうことである．仮にそれが何とか切抜けられても，DATA を出力するときに，実数型のデータに整数型の変換をしようとしていると注意されて，出力できない．双方を切抜けるために，整数型のデータはサブルーチンで整数型から文字型へ変換して，文字型の書式を利用することにした．

2.3 使 用 例

```
CALL OUTREA(6, A)
CALL OUTREA(6, B)
CALL OUTREA(6, C)
CALL OUTLIN(6, 1HO)
```

とすると，A, B, C は1行に横に並んで印刷され，2行目の前後にも CALL OUTLIN (6, 1H0) を入れると，A, B, C は1行に一つずつ3行に印刷される．

```
  DO  1 I=1, 10
1 CALL OUTREA(6, A(I))
  CALL OUTLIN(6, 1HO)
```

とすると，第1行に A(1), ……, A(7) が，第2行に A(8), A(9), A(10) が印刷される．

第 2 表

```
      SUBROUTINE OUTINT(LOGUNO,INTEGR)
C     OUTPUT INTEGER ( LOGICAL UNIT NUMBER, INTEGER )
      COMMON/OUTBUF/NCP,NDI,NFI,DATA(23),FMT(49)
      INTEGER DATA,FMT
      IF(NCP.LT.112)GO TO 1
C     IF THE CHARACTER POSITIONS FOR THE PRESENT DATA ITEM EXCEED THE
C     RIGHT HAND MARGIN, OUTPUT A LINE FOR THE PREVIOUS DATA ITEMS.
      CALL OUTLIN(LOGUNO,1H )
    1 NDI=NDI+1
      DATA(NDI)=IBTOD(INTEGR)
C     BASIC EXTERNAL FUNCTION IBTOD FOR BINARY TO DECIMAL CONVERSION
      FMT(NFI+3)=5HA5,3X
      NFI=NFI+1
      NCP=NCP+8
      RETURN
      END

      SUBROUTINE OUTREA(LOGUNO,SOURCE)
C     OUTPUT REAL NUMBER ( LOGICAL UNIT NUMBER, SOURCE )
      COMMON/OUTBUF/NCP,NDI,NFI,DATA(23),FMT(49)
      INTEGER FMT
      IF(NCP.LT.104)GO TO 1
C     IF THE CHARACTER POSITIONS FOR THE PRESENT DATA ITEM EXCEED THE
C     RIGHT HAND MARGIN, OUTPUT A LINE FOR THE PREVIOUS DATA ITEMS.
      CALL OUTLIN(LOGUNO,1H )
    1 NDI=NDI+1
      DATA(NDI)=SOURCE
      FMT(NFI+3)=5HE12.5
      FMT(NFI+4)=5H.5,4X
      NFI=NFI+2
      NCP=NCP+16
      RETURN
      END

      SUBROUTINE OUTSTR(LOGUNO,STRING)
C     OUTPUT STRING ( LOGICAL UNIT NUMBER, STRING )
      INTEGER STRING
      COMMON/OUTBUF/NCP,NDI,NFI,DATA(23),FMT(49)
      INTEGER DATA,FMT
      IF(NCP.LT.119)GO TO 1
C     IF THE CHARACTER POSITIONS FOR THE PRESENT DATA ITEM EXCEED THE
C     RIGHT HAND MARGIN, OUTPUT A LINE FOR THE PREVIOUS DATA ITEMS.
      CALL OUTLIN(LOGUNO,1H )
    1 NDI=NDI+1
      DATA(NDI)=STRING
      FMT(NFI+3)=5H5A
      NFI=NFI+1
      NCP=NCP+5
      RETURN
      END
```

```
      SUBROUTINE OUTLIN(LOGUNO,CARCON)
C     OUTPUT LINE FEED (LOGICAL UNIT NUMBER, CARRIAGE CONTROL CHARACTER)
      INTEGER CARCON
      COMMON/OUTBUF/NCP,NDI,NFI,DATA(23),FMT(49)
C     NCP = NUMBER OF CHARACTER POSITIONS
C     NDI = NUMBER OF DATA ITEMS
C     NFI = NUMBER OF FORMAT ITEMS
      INTEGER FMT
C     FMT = FORMAT ITEMS
      FMT(1)=5H(   1H
      FMT(2)=CARCON
      FMT(NFI+3)=1H)
      IF(NDI.NE.0)GO TO 1
C     IF THERE IS NO DATA ITEM TO BE PRINTED, WRITE THE CARRIAGE CONTROL
C     CHARACTER ONLY.
      WRITE(LOGUNO,FMT)
      GO TO 2
    1 WRITE(LOGUNO,FMT)(DATA(I),I=1,NDI)
    2 NCP=0
      NDI=0
      NFI=0
      RETURN
      END
```

おわりに

入出力サブルーチンも FORTRAN で書いておけば，ある程度個人の好みに合わせて補修することが容易であり，いわば FORTRAN の入出力がお誂えならば，この入出力は半既製であり，FORTRAN 以外の言語による既製品より便利であるからである．

総合研究「計算機の共同利用」の一環として，東京大学大型計算機センターを利用して進められた．この入出力サブルーチンは同センターのライブラリに登録済である．

いろいろと助言頂いた東京大学工学部森口繁一教授に感謝する．

参考文献

1) IFIP/WG 2.1: Report on Input-Output Procedures for ALGOL 60, Comm. ACM, **7**, 10 (1964), pp. 628-630.
2) Knuth, D.E.: A Proposal for Input-Output Conventions in ALGOL 60, Comm. ACM, **7**, 5 (1964), pp. 273-283.

（昭和42年4月1日受付）