

6805. 二進加法 (続)

一松 信 (立教大学理学部)

二進加法については、今年1月号のプログラムのページに一つのプログラムを発表したが、その後もっとよいプログラムができたので報告したい。前回と同じく FORTRAN での函数サブプログラムになっており、

COMMON IA, IB

でリンクする。

算法はつぎのとおりである。

1. IA と IB をくらべ、 $IA < IB$ なら、いれかえる。 $IA = IB$ なら、答を0としてすぐに戻る。 $IB = 0$ なら、IA を答としてすぐに戻る。
2. 2の累乗 I1 で、 $2 \times I1 > IA \geq I1$ であるものをさがす。
3. IA, IB の一方だけが I1 より大きいか等しいとき、I1 の値を加える。そして IA, IB が $\geq I1$ なら I1 を引く。
4. I1 を半分にする。
5. $IA = IB$ となったらやめる。そうでなければ3に戻って反復する。

反復は $I1 = 1$ になるまでくりかえしてもよいが、 $IA = IB$ になったところで打ち切るほうが早い。 $I1 = 1$

```

FUNCTION NIM(IA,IB)
C      COMMON IA,IB
      NIM = 0
      IF (IA - IB) 900,901,902
900    I1=IA
      IA = IB
      IB = I1
902    IF (IB .NE. 0) GO TO 911
      NIM = IA
      RETURN
911    I1 = 1
912    IF (IA .LT. I1) GO TO 913
      I1 = I1 + I1
      GO TO 912
913    I1=I1/2
      I3 = 0
      I2=IA-I1
      IF (I2 .LT. 0) GO TO 915
      IA = I2
      I3 = I1
915    I2= IB -I1
      IF (I2 .LT. 0) GO TO 916
      IB = I2
      I3 = I1 -I3
916    NIM = NIM+ I3
      IF (IA .NE. IB) GO TO 913
901    RETURN
      END

```

までゆけば、最後ののこりは必ず0となり、 $IA = IB$ の状態になる。

$IA = 0$ (または $IB = 0$) ならば、IB (または IA) を加えて打ち切ることにしてもよいが、この判定は加えていない。

このプログラムは、当初 (これを使用する本番のプログラムとともに) 全国共同利用の東大大型計算機センターで実行を始めたが、現在同センターは、需要に処理能力及ばず、急行でも turn-around time が1週間をこえ、20日かかっても debug が完了しないので、やむなく、最近義父の会社に設置された IBM 360/44 で実行してもらった。特殊記号のコードの差 (26型と29型) のために、ちょっとしたトラブルがあったが、30分たらずで (これを利用した本番プログラムをも含めて debug と) 完了した。

その後ひきつづき、これを利用して、各種の石とりゲームの必勝法の計算を実行しているが、これらはあまりにも特殊な話題であるから、あらためて別の形で発表したいと思う。

6806. 単純な読み書き

西村 恕彦 (通商産業省電気試験所)

入力ファイルから入力レコード (さん孔カード) を読み込んで、そのまま出力ファイル (行印刷機) に出力レコードとして書き出す。これを入力ファイルがファイルの終りになるまで繰り返す。

このコボル語のプログラムの目的は、データカードの内容を印刷することと、はじめて使う計算組織の性質をたしかめることとである。後者も実地の経験から発生した要求であって、以下の諸点を確認できる。

1. モニター—コンパイラ—ロードの機能が正しく働いていること。
2. 計算室の操作員のコンベンション。
3. コントロールカードのつけかた。
4. 入出力装置の指定のしかた、そのほか特定の計算組織に固有なコンベンション。
5. ソースプログラムのさん孔—印文字号系。
6. データのさん孔—印文字号系。

これらはいずれも、大きなプログラムをいきなり動かしたのでは、いろいろな誤りと複合して、はっきり確認することの困難な点である。このうち、第3項以降の諸点は、初歩的な解説書に当然記載されているはずのものが、実際にはなかなか書いてない。

このプログラムは CODASYL COBOL の文法に厳密にしたがって書いてあると同時に、少なくとも一つ