きはフローに直してみる。すると論理ミスがチェックされる。そんなとき，その方はフローを書いてない場合が多い。

この「流れ図」になじませるにはカウントのとり方最大値，スウィッチ等の例題によってひとつのパターン（型）を定着させることも意義がある。または，フローからプログラムを書かせたり，プログラムをフローに直す例題も役立つ。

フローの必要性を，コンピュータの経験者に言わせれば即コーディング即パンチ即ランと同等の価値があるという。語気を弱めても，フローを書かない方はコーディングの意味を半解しているので，プログラミングにのびがないという。

経験者はプログラミングの段階で，フローを書いた方が適確であることをじゅう分体験させられている。従ってフロー的思考を基盤とする態度を指導する必要のあることを語ろうとするのであろう。

換言すればプログラムは「思考して」作るものだからA面のもっている約束事よりは比重が重いともいえよう。

私としてはB面通りにコンピュータが動いたときの喜びの方を与えたい。コンピュータの約束にふりまわされるよりは，約束をつかってコンピュータを利用する人間を育てたいからである。

—C面—

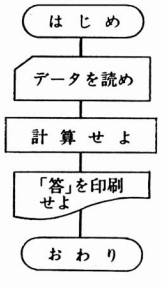
フォートラン文法では，例えば欄記述子をどこまで指導したらよいものか—という観点も捨てられない—面である。

それはIタイプでせん孔してあるデータをFタイプで読ませるとか，スラッシュ1本で「行」がかわる機能とか，文字印刷をさせたいとか……という指導欲の派生をともなっているのでA面の選択のときよりは調整時間をくうものである。

私としては，はじめからこまごまと説明するのは混迷させることもあるので，あまり当初から欲張った内容を盛り込まない方がよいと思う。

コンピュータの指導は「コンピュータをこわがらせないようにすることから始まる」ものなので，いきなり約束をごたごた披露するのはさけたい。

下図はプログラムの最も基本とされているものである。



時間がなくとも，この基本プログラムにはじゅう分に時間をかけたいものである。

このなかでも算術代入文での変数およびその取り扱いとか，リードフォーマットでのデータとの対応には更に時間をかける必要がある。

極端な話しだが，この基本プロの指導がゆきとどいていれば，立派にプログラムというものの概念を教え得たことになる。なお，これからのプログラムに取組む姿勢がととのったといえよう。

即ち，最低のフォートラン言語によって，とも角コンピュータが動いたことを体験させておくと，あとで派生する約束事の理解をはやめるものである。

浅学なのでA面，B面，C面位しか観点をとらえ得なかった。

実際の比重は2：3：1である等とはいえない。ステップごとに比重が変化するからである。あるステップでは0：10：0のときもある。

またB面にこだわって組み込み関数を忘れてしまうと，逆にプログラムの能率をおとす結果ともなる。あるいはC面をおろそかにすると思わぬ誤差の発生をみて困惑を招くこともある。

そして，A，B，Cのほかに，折にふれてハード的な説明を加えることも考えておかなければならない。

しかしここでは指導内容のとらえ方としてはA面，B面，C面等というもの

すなわち

1. 基本的ステートメントとしてなにを採用するか。
2. フロー的思考態度をどのようなステップでととのえていけばよいか。
3. コーディング・テクニックの指導に適した問題はなにか。

という立場をつねに留意しておく必要のあることだけを述べておきたい。

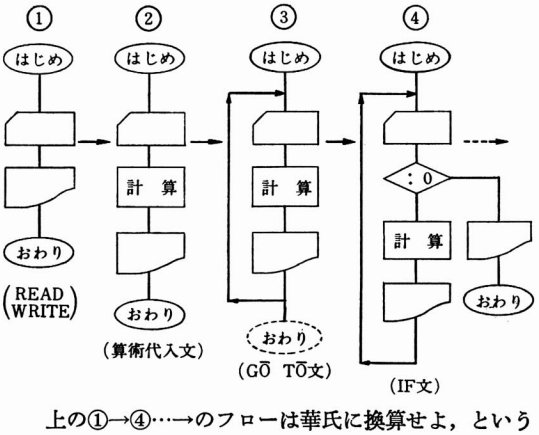
・　・　・

(2) 指導方法について（一例）

どこから導入するか等というような十人十色的なことにはふれない。

ここでは「ステップ方式」にとどめておきたい。

—ステップ方式—



上の①→④…→のフローは華氏に換算せよ，という問題を「ステップ方式」によって順序書きしたものである。

ステップ方式というものは，せん孔（紙カードとす