

プログラミングの実習におけるマークセンスの応用*

小林 光 夫**

1. はしがき

計算機の普及にともない、その使用者を大量に養成することが現下の大きな課題の一つとなっている。使用者養成のための初心者教育には、初心者自身によるプログラムの作成および計算機によるその処理を行なうところのいわゆる実習が、初心者に計算機を身近に感じさせる意味からも、また教育過程において初心者の犯す誤りにすばやく適切なフィードバックをかける意味からも、きわめて重要である。

さて、大量の初心者に実習を施す際には、パンチの大量処理を行なわねばならぬが、せん孔機が多数使用可能な場合は別として、通常は、いかに早く大量のせん孔をこなすかが一つの問題となろう。この解決への試みとして、マークセンスおよびポータパンチを用いた例がアメリカにある¹⁾。

本報告では、われわれが行なったマークセンスによるパンチの大量一括処理の実験的な試みを簡単に述べ、今後の方針等を検討してみたい。

2. 目的および方法

FORTRAN IV の初心者講習会の実習において、マークセンス方式を用いて、人間の手によらない自動的なせん孔を行ない、大量のプログラムやデータの一括処理を行なうことが目的である。

われわれの用いたマークセンス用のカード（以後Mカードと略記する）は第1図に示したデザインのものである。このMカードにマークすることによって、FORTRAN IV によるプログラムおよびデータを作成するのである。Mカードへのプログラムやデータの書き方などについての詳細は、文献2)を参照されたい。

使用した計算機（HITAC 5020）が、カードを入力

* An Application of Mark Sensing Technique to the Exercise of Programming, by Mituo Kobayasi (Department of Mathematical Engineering and Instrumentation Physics, Faculty of Engineering, University of Tokyo)

** 東京大学工学部計数工学科

とし、FORTRAN IV (HARP 5020) のソースデックを連続的にモニタ処理することが可能であったため、実習の実施にあたっては以下に示すような方法をとった（第2図を参照）。

(1) まず、FORTRAN 言語によりプログラミングを行ない、コーディングシート上にプログラムとデータを書く。

(2) つぎに、(1)で書かれたプログラムとデータをMカードに導電性の強い鉛筆で書きうつす。

(3) (2)でマークされたMカードは、各人を区別するために特別に用意されたMカードを先頭に添え、一括して機械（IBM 519）にかけて自動せん孔を行なう。

(4) (3)でせん孔されたMカードを入力データとし、計算機のせん孔出力機構を利用して、FORTRAN IV のソースデックを作成する。せん孔作成されたソースデックには、一括してモニタ処理が可能となるように、各人のデックの先頭と末尾に制御カードが自動的に入るようになっている。制御カードには、あらかじめ打切計算時間と打切印刷行数の指定が入るようにしてあり、無駄な計算を長時間続ける危険性をふせいでいる。

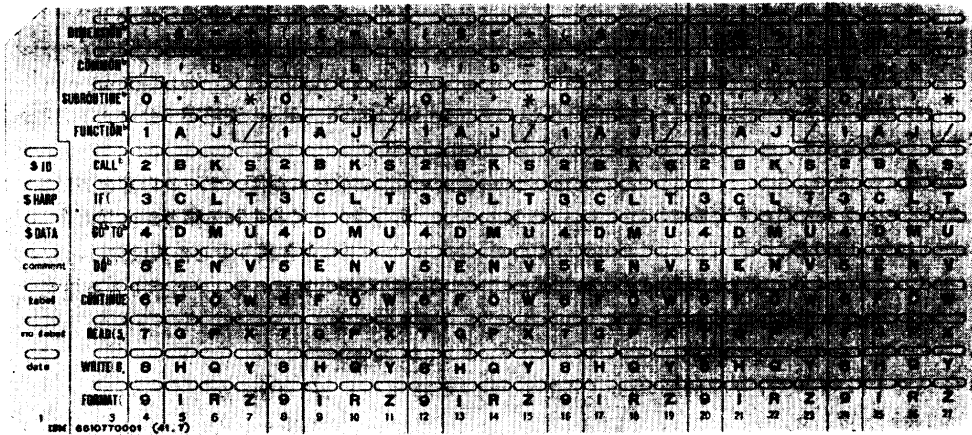
(5) (4)でできたソースデックを一括してモニタ処理する。

(6) (4)でできたカードには、印字がなされていないので、初心者の理解を助けるために、カードをインタープリタにかけて印字を行なう。

(7) (3)~(6)で一括処理したMカード、ソースデックを各人別に仕分けし、(5)で得られた計算結果とともに各人に返却する。

3. 実施結果

昭和41年9月2日（金）、3日（土）および5日（月）に開かれた日本化学会関東支部主催の電子計算機講習会において、マークセンス方式を用いた実習を行なう機会を得た。実習は2日と3日の両日、午後1時から3時まで行なわれ、計算結果は2日の実習の分



マークセンス用カード (M カード) の書き方

コーディングシートの1行 (continuation) で一つのステートメントが2行以上にあつたときには、便宜上1行とみなすことにする) を M カード数枚に書きうつす。その際、M カードの第1列で、コーディングシートの行の種別を示し、引きつづく列で左から右の順にステートメント、コメント、データなどを書く。M カード1枚で書ききれないときには、続けて何枚の M カードを使ってもよい。ただし、続く M カードの第1列はマークしてはならない。

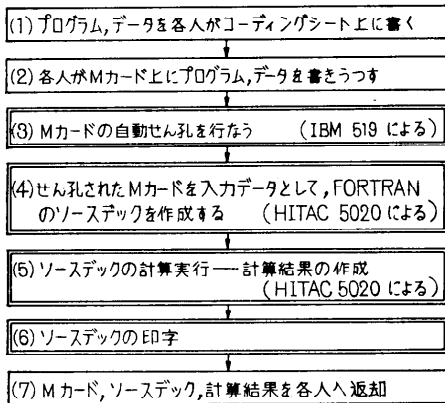
M カード 第1列の説明

- \$ ID: 制御用の特別な M カード、引きつづく列に氏名、番号などを書く。
- \$ HARP: 引きつづく列にチェック名を書く。
- \$ DATA: " " データを書く。
- comment: " " コメントの内容を書く。
- label: statement label のあるステートメント、引きつづく列に statement number を書き、次の M カードからステートメントの内容を書く。
- no label: statement label のないステートメント、引きつづく列にステートメントの内容を書く。
- data: 引きつづく列にデータを書く。

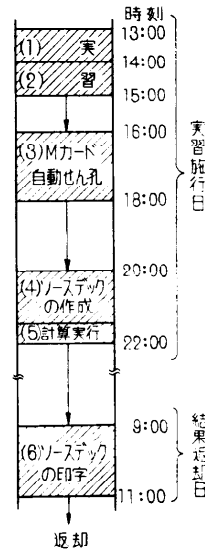
注 意

- (1) 空白は必ずbのところにマークすること。
- (2) no label のマークは忘れやすいから、気を付けること。
- (3) 1列に二つ以上のマークがあつてはならない。

第1図 マークセンス用のカード (M カード)



(二重枠内は一括処理)
第2図 処理の手順



第3図 スケジュール

は3日の午前中に、3日の分は5日の午前中に返却された。前節に述べた方法を実施するにあたって予定したスケジュールは2回の実習とも第3図のとおりであ

る。なお受講者は127名であった。

使用計算機の都合などで、実際にはこのスケジュールから多少のずれはあったが、各処理に要した時間はほぼ予定のわく内におさまった。以下に実際の処理結果に関するデータを示す。

第1回実習

- 124件のプログラムとデータを処理。
- Mカードは約3500枚。
- Mカードの自動せん孔に要した時間は約1時間弱。処理を依頼したため正確な所要時間は不明。
- Mカードからソースデックの作成：
 - 19時15分～21時30分、約2時間15分（予想は1.5～2時間）要した。
 - せん孔作成されたソースデックは約2500枚。
 - Mカードに二重あるいは三重せん孔のものがあ
り、カード読み取りが不可能となり、計算機がス
トップする事態がいく度か生じた。このため処理
時間が予想以上にかかったものと思われる。
- ソースデックの計算実行：
 - 21時30分～22時00分、約30分。
 - Mカードの書き誤りが多く（書き方の教育が悪か
ったためと思われる）エラー続出。計算結果の得
られたプログラムは17件にすぎない。
- ソースデックの印字：
 - 9時45分～10時45分、約1時間。
- 第1回実習において気づいたエラーで、Mカード
の書き誤りによると思われるもの：
 - われわれの用いたMカードは一つの列には唯一
つのマークをすることしか許されないが、同一列
に二つ以上のマークをする誤りがあった。これに
よってMカードが二重あるいは三重せん孔され、
せん孔された穴の位置が、計算機がデータとして
読み取り不可能なコードであるとき、計算機がス
トップしてしまう事態が生じる。
 - Mカード第1列のマークの仕方間違い。われわ
れはMカードの第1列によって、コーディング
シートに書かれたプログラムあるいはデータの1
行の性格づけを行なったのであるが、これが初心
者には難かしく感じられた。
 - 制御カードに相当する特別のMカードを初心者
自身にマークさせたため、このMカードと普通
のプログラムを書くMカードとの混同があった。
 - マークの仕方の悪いもの、例えばマークのうすい

もの、導電性の弱い普通の鉛筆でマークしたもの
などがあった。普通の鉛筆では濃くマークしても
正しくせん孔されない場合がある。

○FORTRAN 言語の理解不十分によると思われるエ
ラー：

- 実数と整数の混同、データの書き方間違いなどの
初心者によくみられる誤りが非常に多い。
- Mカードの書き誤りをした人は、ほとんどの場合、
FORTRAN 言語の文法的な誤りを併発している。

第2回実習

- 129件処理。
- Mカードは約4000枚。
- Mカードの自動せん孔：
 - 1時間以内で処理完了。
 - 自動せん孔の際のミスなし。
- Mカードからソースデックの作成：
 - 22時05分～22時49分、所要時間44分。
計算機の都合で処理開始時刻が遅れた。
 - せん孔作成されたソースデックは約3000枚。
- ソースデックの計算実行：
 - 22時55分～23時30分、所要時間30分。
 - 計算結果の得られたプログラムは86件。
約7割が計算機を通ったことになる。
- ソースデックの印字：
 - 9時30分～10時40分、所要時間1時間10分。
- 第2回実習において気づいたエラーで、Mカード
の書き誤りによると思われるもの：
 - Mカード第1列の書き誤りによるもの少数。
 - 他は、Mカードのデザイン上の欠点によると思
われるエラー、たとえば `FORMAT(...)` の右か
っこ ')' を忘れる、' ,' と ' ' ' を混同する、
' 0 ' と ' 0 ' を混同するものなど数件みられた。
- FORTRAN 言語の理解不十分によると思われるエ
ラーは、第1回実習のときにみられたエラーと同程
度の初心者が犯しやすいエラーがわずかにみられた
程度である。

4. 諸問題点とその解決策

実習におけるマークセンスの応用の長所、短所をあ
げてみると、長所は、人間によるカードパンチの手間
がはぶけること、大量のプログラムやデータを一括処
理できること、いったんプログラムやデータがMカ
ードに書かれてしまえば、あとは機械的に処理可能で

あるから、パンチミスのようなプログラマ自身以外に原因のあるミスが少なくなることなどであろう。

他方、短所は、プログラムやデータを M カードに書きなおす手間がかかり、この作業が初心者にとってはかなりむずかしく、この段階でのエラーが出やすいこと、1ジョブについてのソースデッキの枚数に比べて、それに必要な M カードの枚数が多くなること、現在の方法ではプログラムやデータの修正を行なうには、1人分の M カードを全部書きなおさねばならぬことなどであろう。

ところで、このマークセンスの応用の実験的な試みで気のついたことは、1回目の実習では M カードに起因すると思われるエラーが非常に多く、計算結果の得られたのはわずかであるが、2回目の実習では約7割の好成績をおさめている点である。このことは、M カードの書き方の教育法を工夫すること、M カード上の印刷のデザインに改良を加えることなどによって、マークセンス方式の欠点をかなりカバーできることを示唆しているように思われる。以下に、前掲の M カードに起因するエラーに対して、いくつかの対処案を述べてみる。

(1) 同一列に二つ以上のマークをする——教育によってきりぬける。あとは、M カードの自動せん孔を行なうときに、二重あるいは三重せん孔のデバッグをする可能性を検討する。

(2) 第1列の書き誤り——コーディングシートに書かれた行の性格づけを初心者に行なわせることは、かなりむずかしい知的作業と思われるので、1行を M カード数枚に書きうつしたときにエンドマークを添えてマークするなどの工夫をするのが良いと思われる。

(3) 制御カードに相当する M カードと普通のプログラムを書く M カードとの混同——特別の M カードは初心者マークさせず、あらかじめ初心者の氏名、あるいは番号を印字せん孔したカードを渡しておき、プログラムの先頭に付けてもらう。

(4) FORMAT (...) の右かっこ ')' 忘れ、',' と '' 混同、'O' と '0' の混同など——M カード上の印刷のデザインの変更、あるいは教育を厳格に行なう。

今回用いた方法では、M カードからいったん FORTRAN のソースデッキをせん孔作成しているが、これは将来は、磁気テープあるいは磁気ドラムなどにファイルすべきであろう。このようにすれば、処理の時間的な損失も少なくなるであろうし、プログラムやデータの修正も、修正カードのような特別の制御用 M カードを添えることによって、個人別のファイルの一部に修正を加えるようにできるから、現在の一点である一人分の M カードを全部書きかえることも必要でなくなるであろう。

5. む す び

種々の高度な計算機技術の発達している今日、マークセンスの応用はいささか古めかしく思われるかも知れないが、プログラミングの大量教育をこなす一手段として試みしてみる価値はあるように思われる。今後、さらに改良を加えて試みる機会が得られたら幸いである。この試みにあたって、東京大学の森口繁一教授には、終始懇切なご指導をいただいた。日本化学会関東支部の方々には、プログラミングの講習会開催に尽力していただき、また鹿島建設技術研究所の計算センターの方々には、計算機使用の便を計っていただいた。記して感謝の意を表します。

参 考 文 献

- 1) W.F. Luebbert: Mark Sense and Port-A-Punch Programming Inputs, CACM, Vol. 7, No. 12 (1964), p. 722.
- 2) 小林光夫: プログラミングの初心者教育におけるマークセンシングの応用, 第8回プログラミングシンポジウム報告集, 1967, B 58~B 72 ページ.

(昭和42年4月1日受付)