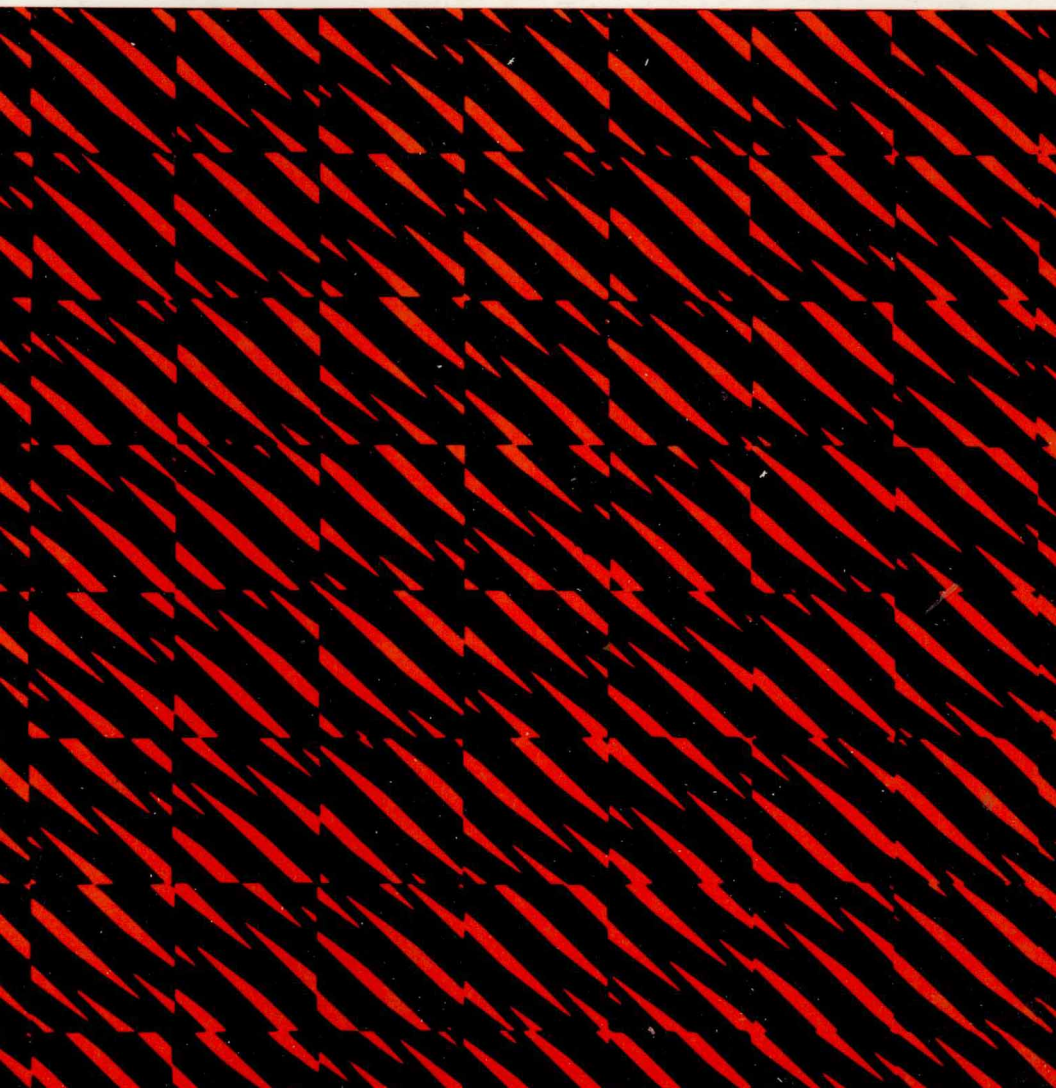


FORTRAN による

プログラミング入門

細井 勉、嶋田君枝著

産業図書



FORTRANによる

プログラミング入門

細井 勉 嶋田君枝著

産業図書

<著者略歴>

^{ほそ} 細 ^い 井 ^{つとむ} 勉

昭和 39 年 東京大学理学部数学科卒業
 昭和 41 年 東京大学助手
 昭和 45 年 理学博士
 昭和 46 年 津田塾大学助教授
 昭和 48 年 津田塾大学教授
 昭和 53 年 東京理科大学教授 (理工学部)
 現在にいたる

^{しま} 嶋 ^だ 田 ^{きみ} 君 ^え 枝

昭和 46 年 津田塾大学数学科卒業
 昭和 48 年 津田塾大学助手
 昭和 54 年 東京理科大学助手 (理工学部)
 現在にいたる

FORTRANによる プログラミング入門 定価 1800 円

昭和 55 年 1 月 7 日	初 版	
昭和 55 年 5 月 10 日	第 2 刷	
	著 者	細 井 勉 嶋 田 君 枝
	発 行 者	森 田 勝 久
	発 行 所	産業図書株式会社 東京都千代田区外神田 1-4-21 郵便番号 101-91 電 話 東京 (253) 7821(代) 振替口座 東京 2-27724番

© Tsutomu Hosoi 1980
 Kimie Shimada

東陽印刷・清水製本

まえがき

コンピュータって、巨人かもしれない、怪物かもしれない。だけど、コンピュータなんておもちゃなんだ、と思う人がいてもよいのではないか。おもちゃでも扱うように、気軽にコンピュータを使う人がふえてくれるといいな。コンピュータと遊ぶのを楽しむ人がふえてくれるといいな。

こういう気持ちで、こういう願いをこめて、この本を書きました。

プログラミングの手ほどきからはじめました。そして、ちょっとした遊びをしたりするのに必要な程度の応用プログラミングの基礎部分まで、できるだけやさしく、わかりやすく解説したつもりです。いろいろな実的な練習問題もたくさん用意しました。また、わたくしたち自身が楽しんだ経験的なお話も入れておきました。

最初に述べた気持ちと願いは、ずっと前から、わたくしたちがもっていたものです。こういう気持ちをもって、プログラミングの入門教育やその上の段階の教育にあたってきました。そして、そのような入門時の教育においてエッセンスとなるものは何かということについて、わたくしたちなりに理解が深まったので、この本を書きました。

コンピュータと遊びたい人たちにとって、すぐに役立つような本が少ないので、このような参考書が必要ではないかと思ったことも、この本を書いた動機の一つです。

内容は3部にわけました。

第1部では、FORTRAN によるプログラミングの手ほどきをしてあります。

す。FORTRAN についてこれだけ知っていればどんな計算でもできる、という最低範囲を少しだけ超えた程度の道具立てにしています。プログラミングの入門書であり、くわしい文法書を学ぶ準備的な本でもあるようにしたつもりです。これだけの知識でどんなプログラムも書けます。十分にコンピュータを使いこなせます。

第2部では、応用的なプログラミング技法を述べました。“データ構造”とそれを操作する“アルゴリズム”に重点をおいて解説しています。内容的には、すこし程度の高いところまで立ち入っています。

ここで紹介するアルゴリズムは、FORTRAN によらなければならないというわけではありません。アルゴリズムを述べるのに、何らかの言語が必要です。そのために、たまたま、FORTRAN を利用しただけのことです。そして、ここで FORTRAN を利用するため、第1部で FORTRAN を紹介したのです。

第3部は、おしゃべりです。息ぬきのページでもありましょう。独立した読み物です。ここから読みはじめてもかまいません。

全体を通して、例題でも、練習問題でも、パズル的な材料をたくさん取りあげました。パズルなどをコンピュータで扱ってみたい人たちの手助けになりたいと思ったからです。でも、遊びを助けるためだけにこの本を書いたわけではありません。ふつうの入門教育でも、遊びの精神をもつことは、“ゆとり”の教育として大切なことだと思うのです。また、ここで解説した技法は、遊びのためだけでなく、いろいろな分野でのプログラミングにとって欠かせない重要なものだと思うのです。

この本で紹介しているアルゴリズムや具体的なプログラムは、かならずしも、洗練されたものではありません。どちらかという、粗雑なものです。また、“よい”プログラムを書くための心得とかマナーとかにもふれていません。端的にアルゴリズムを紹介したかったからです。そして、コンピュータにはやくなれてもらいたかったからです。この本のプログラムの粗雑さが気になる方は、もっと改良してみてください。これも、また、練習問題です。

全体の構成は二人で打ち合わせしましたが、執筆は、第1部は細井が担当し、第2部は嶋田が担当しました。第3部の最初の三つの章は細井が執筆しました。

が、後半の三つの章は嶋田が以前に雑誌に発表したものを再掲しました*。

最後になりましたが、このような本を出版する機会を与えて下さり、また、執筆に際していろいろと相談にのって下さった、産業図書の江面竹彦氏に感謝いたします。また、印刷にあたってお世話になった産業図書の鈴木正昭氏、西川宏氏、日吉育子氏にも感謝いたします。

昭和54年11月

細 井 勉
嶋 田 君 枝

* 4×4 魔方陣の話，数学セミナー 1973 年 2 月号，組木パズル（前・後編），数理科学 1976 年 7 月号，8 月号。

目 次

まえがき

第1部 プログラミングの基礎

1. コンピュータを動かすのは.....	3
計算機構, プログラム, アルゴリズム	
2. はじめはタシ算から.....	9
算術式, READ, WRITE, FORMAT, STOP, END, INTEGER	
3. “もしも” できるわかれ道.....	17
IF, GOTO, 論理式	
4. 数もいろいろ	23
実数型と整数型, REAL, 基本外部関数, 組込み関数, 2分法	
5. ぐるぐる回せ	31
DO, CONTINUE, 配列, DIMENSION	
6. コトバでごあいさつを	37
文字の入出力, DATA	
7. 行列は何次元でも	45
多次元配列, 文関数, 引数	
8. 呼べば答える副プログラム	53
関数副プログラム, サブルーチン副プログラム, FUNCTION, SUBROUTINE, CALL, RETURN, COMMON	
9. 最後に整列を.....	61
ソート, 選択法, 交換法, 番地計算法, 挿入法, マージ	

第2部 応用プログラミング

10.	一つ、二つ、三つ、四つ	69
	枚挙法, 場合の数	
11.	9個を3個ずつに	89
	順列, 組合せ, 集合の分解	
12.	木, 林, 森	99
	木生成, 2分木, 木探索	
13.	ふえたり, へったり, つみあげたり	113
	スタック, ポインタ管理	
14.	あっちへ行ったり, こっちへ来たり	121
	リスト, グラフ	
15.	出口, 入口, 出ないで入る	137
	再帰呼出し, リエントラント	
16.	大きい数, 長い数	145
	多倍長表現, 多倍長計算	

第3部 いろいろな話題

大きな素数の話	155
3山くずしの必勝法	159
数のお遊び	163
4×4魔方陣の話	171
組木パズル (前編)	181
組木パズル (後編)	193
索引	205

第1部 プログラミングの基礎

コンピュータに仕事をやってもらうためには、まず、“プログラム”というものを用意しなければなりません。そして、プログラムは、コンピュータに理解できる特別な“言語”で書かねばなりません。これはもうよく知られていることでありましょう。

第1部では、そのプログラムの書き方の基礎、つまり、プログラミングに必要な最小限の道具を紹介しましょう。

ところで、プログラミングのための言語にはいろいろな種類があります。ここでは、それらの中から、FORTRAN(フォートラン)とよばれている言語を紹介します。

FORTAN を取り上げるのは、それが優れていると思ったからでも、それがこの本で扱うプログラミングに適していると判断したからでもありません。言語は、じつは、何でもよかったのです。というわけは、この本では、プログラムを書く際の準備的な考え方を解説したいのです。その考え方は基本的なものですから、原則的に、たいいていの言語に適用できるものなのです。それで、非常に広く使われている FORTRAN を取り上げることにしただけなのです。

コンピュータを動かすのは

計算機構、プログラム、アルゴリズム

コンピュータを物理的に動かしているのは電力でしょうが、論理的には、プログラムがコンピュータを動かしています。

コンピュータは計算機械です。でも、いつも同じ計算を繰り返しているわけではありません。コンピュータは多種多様な計算のできる機械ですし、実際、いろいろな計算をしています。コンピュータがどういう計算をするかは、“プログラム”として用意された“計算手順”の指示によって定まります。プログラムを取りかえることによって、違った計算がなされることになります。

プログラムがコンピュータを“論理的に”動かしている仕組みを理解するためには、コンピュータの構成についてのおおまかな知識が必要です。

ですから、まず最初に、コンピュータの構成についての原理的な話からはじめましょう。

コンピュータの中では、数値データにせよ、文字や記号を並べたプログラムにせよ、あるいは、内部の制御用の信号にせよ、すべて、2進数、つまり、0と1を並べたもので表わされ、扱われています。数の正負の符号も0や1で表現されているわけです。

2進数で表わされたデータの長さ、つまり、2進数としての桁数をビットという単位で数えます。これは、論理的に考えると、情報の量を測る最小単位です。

一般的には、ビットという単位は小さすぎるので、何ビットかをまとめた単

位が使われます。

まず、8ビットを1バイトといいます。ふつう、文字や信号は、8ビット、つまり1バイトの2進数に変換されて扱われます。1バイトで表わされうる2進数の種類数は $2^8 (=256)$ です。

バイトの上の単位に語という単位があります。2バイトを1語とするコンピュータもありますが、ふつうの汎用コンピュータでは、多くの場合、4バイト、つまり32ビットを1語と数えます。このとき、2バイトは半語、8バイトは2語となるわけです。数値データは1語あるいは2語の2進数に変換されて扱われます。1語で表わされうる2進数の種類数は $2^{32} (=4294967296)$ です。 $2^{10} (=1024)$ 語をまとめて1k語(1キロ語)といいます。

コンピュータには、ふつう、数k語から数千k語あるいはそれ以上の情報をしまっておける記憶装置が用意されています。記憶装置には、かなりに高速に内容を出し入れできる内部記憶装置と、出し入れが遅い外部記憶装置とがありますが、ここでは区別しないことにしておきます。

記憶装置の内部では、語単位(あるいは、バイト単位)に一連番号がつけられていて、その番号をその語(あるいは、バイト)の番地とよびます。データの出し入れは番地指定によります。つまり、100番地にデータをしまおうとか、200番地からデータを取り出すとかいうわけです。

さて、コンピュータは一つ一つの加算とか除算とかの演算を積み重ねて、“計算”を完成させます。

個々の演算を実行する部分は演算装置です。ここにはいくつかのレジスタとよばれる一種の記憶(と演算)の場所が用意されています。そして、指定されたレジスタの中のデータや、記憶装置の中の指定された番地のデータなどの間で、指定された加算とか除算とかの演算がなされます。

コンピュータの動作全体を制御する部分は制御装置です。この部分は、演算装置にいろいろな指定を送って個々の演算を完成させるだけでなく、一連の演算がつぎつぎとなされるような制御も行ないます。

計算を構成する演算の実行順序、つまり、計算の手順は、まず、プログラムとして準備され、それが2進数に直されて記憶装置の中に入れておかれます。制御装置は、記憶装置から、原則的には番地の順に、プログラムの一部分とし

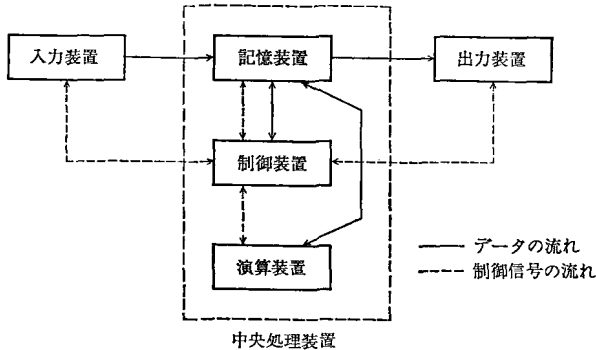


図 1.1

ての2進数を取り出しては、その内容に応じて演算などのための制御信号をコンピュータの各部分に送って、計算を完成させるのです。

コンピュータには、さらに、外界とデータのやりとりをするための入力装置と出力装置がつながっています。入力装置からは外部のデータが2進数に変換されて取りこまれます。出力装置に送り出された2進数は、それぞれの装置によって定まっている変換規則にしたがって、文字なり記号なりに直されて、紙の上の印字となったり、ブラウン管の上の文字となったりします。

制御装置はこれらの入出力装置も制御します。

さて、プログラムは、ふつう、何らかの言語で紙の上などに書いて用意します。この段階のプログラムを原始プログラムといい、プログラムを記述するための言語をコンパイラ言語といいます。この本で紹介する FORTRAN はコンパイラ言語の一種です。

つぎに、原始プログラムは、コンピュータに読みこんでもらうために、カードとかテープとかに穿孔（パンチ）されます。そして、そのカードやテープを入力装置にセットして、コンピュータに読みこんでもらいます。場合によっては、カードやテープを使わないで、別の方法でコンピュータに読みこませることもあります。読みこみは、コンピュータの中に用意された“コンパイラ”とよばれるプログラムの実行によってなされます。読みこまれるとき、原始プログラムは、機械語という制御装置が理解できる形の2進数に翻訳されて、記憶

装置の中に格納されます。この形のプログラムを目的プログラムといいます。そして、このプロセスを“コンパイルする”といいます。

こうして作られた目的プログラムを計算手順だとみなして制御装置が実行するとき、計算がなされるわけです。

プログラムに限りませんが、一般に、物事の実行手順のことをアルゴリズムといいます。より正確には、有限回のステップで終了する手順だけをアルゴリズムといいます。

この本の目的は、コンピュータのためのいろいろなアルゴリズムを紹介することです。

練習問題 1

1. k は 2 以上の整数とします ($k \geq 2$)。そのとき、1 以上のどんな整数 x に対して、つぎの条件を満たすような正整数 n と負でない整数 $a_n, a_{n-1}, \dots, a_1, a_0$ がただ一通り定まります。

$$\begin{cases} x = a_n k^n + a_{n-1} k^{n-1} + \dots + a_1 k + a_0 \\ i = 0, 1, \dots, n-1 \text{ に対して, } k-1 \geq a_i \geq 0 \\ k-1 \geq a_n \geq 1 \end{cases}$$

このとき、 $a_n, a_{n-1}, \dots, a_1, a_0$ をその順序に並べた表現

$$a_n a_{n-1} \dots a_1 a_0$$

を x の k 進法表現といいます。

問 1 つぎのアルゴリズムで $n, a_n, \dots, a_0$ が求められることを証明しなさい。

ステップ 1 $i=0$ とおく。(ステップ 2 に行く。)

ステップ 2 x を整数の範囲で k で割り、余りを a_i とし、商をあらたに x とおく。(ステップ 3 に行く。)

ステップ 3 $x=0$ ならステップ 4 に行く。 $x \neq 0$ なら、 i を 1 ふやしてステップ 2 にもどる。

ステップ 4 $n=i$ において、終り。

問 2 10 進法表現 524287 を 2 進法表現に直しなさい。また、16 進法表現にも直しなさい。ただし、16 進法表現の場合、 a_i が 10, 11, 12, 13, 14 あるいは 15 になったら、それらをそれぞれ A, B, C, D, E あるいは F という文字で表わしなさい。

問 3 2 進法表現 11110111100 を、10 進法表現と 16 進法表現とに直しなさい。また、16 進法表現 FACE を 2 進法表現と 10 進法表現とに直しなさい。

2. つぎのような 5 枚のカードがあります。

A

16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31

B

8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31

C

4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31

D

2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31

E

1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31

さて、1から31までの数を一つ心に思ってください。その数がどのカードに入っているか、調べてみてください。AとCとDに入っているとしましょう。そのとき、これら3枚のカードのトップに書いてある数を加えてみましょう。16+4+2=22。それがあなたの思っていた数です。

問1 カードの仕組みがどうなっているか、2進法表現と関連づけて考えてみなさい。

問2 1から63までの数をあてることができるような、6枚のカードの作り方を考えてみなさい。

3. 天秤と6個の分銅を使って1g(グラム)きざみで重さを計ることにします。

問1 分銅と計量物はいつでも別の皿にのせることにしたとき、63gまで計れるようにできます。どのような重さの分銅を用意するのでしょうか。

問2 計量物をのせる皿にも分銅をのせてよいことにしたとき、364gまで計れるようになります。どのような重さの分銅を用意するのでしょうか。

問3 問2と3進法表現との関係を考えてみなさい。

4. 0と1から成る3ビットの(2進)数字列は8通りあります。その全部をつぎのように並べてみます。

000, 001, 011, 010, 110, 111, 101, 100.

この列は、隣りに移るとき、ただ1箇所の桁だけで0と1がかわるようになっています。さらに、最後の100から最初の000に移る際にも、この性質が成り立っています。2進数字列をこのように並べたものを(上の場合には3ビットの)交番符号列といいます。

問1 5ビットの交番符号列を作りなさい。

問2 一般に、 n は与えられた正整数であるとき、 n ビットの交番符号列を生成するアルゴリズムを示しなさい。

はじめはタシ算から

算術式, READ, WRITE, FORMAT, STOP, END, INTEGER

さあ, FORTRAN という言語の紹介をはじめましょう.

各章ですこしずつ解説します. それぞれの章の範囲で, それ以後の章の知識を使わないで, 何らかの完全なプログラムが書けるようにしてあります. ですから, 後でまとめて実習をしようとしたりはしないで, 各章の練習問題を使ってこまめにプログラミングの実習を行なってみてください.

コンピュータを実際に使用できる場合には, 努力して, 自分のプログラムをコンピュータにかけてみてください.

コンピュータはよい教師です. 思い違いをそのまま実行するとどうなるか示してくれることがあります. 思い違いを発見してくれることもあります. 間違うことを恥じないで実習してみましょう.

実習こそ, 習得の早道なのです.

本書で(第1部で)説明する FORTRAN 文法は, ごく一部分の限られた範囲のものです. もっと便利な機能を知りたい場合には, 第1部を理解した上で, 本格的な文法書について学ぶとよいと思います. ただ, 本書の範囲で, たいのプログラムが書けるようにしてあることだけは, 注意しておきます.

本書では, 例を学ぶことによって知識を増すようにしてあります. “真似”をしてください. そのために, 禁止的な説明はほとんどしていません. “してはいけないこと”が気になる場合にも, 本格的な文法書を参照してみてください.