

「情報処理」1年文I/IIクラス9-10 #0

久野 靖*

1994.10.17

0 はじめに

「情報処理」は昨年度から全員必修として実施されるようになった科目で、情報教育南棟の完成に伴い今学期から Unix システムによる実施となった…そうです。(非常勤講師なのでよく知らない。) クラスごとの内容や進め方は原則として担当教官に任されるということなので、私としては次のような目標を立てさせていただきます。

- a. 計算機システムの原理と構造を理解する。
- b. 計算機を道具として使いこなせるようになる。
- c. アルゴリズムとプログラミングについて理解する。

これらは一応、互いに「直交」しています。例えば計算機の原理もプログラムの作り方も知らないままワープロソフトで文書を作っている人はたくさんいますし、逆にアルゴリズムの研究だけが趣味でそれ以外では一切計算機を使わない人もいるかも知れません。皆様の多くはbの「使い方」だけ教われれば十分でaもcもパスしたいと思っただけかも知れませんが、そこは勉強しなくても勉強させられるのが必修科目ですからそのつもりでいてください。でも理解してしまえばaやcの方がずっと面白いです(と、計算機屋の私がいっても説得力はないですね)。

次に、科目の運用について説明させていただきます。まずこのクラスは毎週月曜日の14:40~16:10(90分)ということですが、その中で上に挙げたa~cがバランスするように各回の内容を調整していくつもりです。なお、次回からはこの科目はすべて情報処理棟の大演習室で実施されます。このことからわかる通り、実習は非常に重視されます。今週はまだ南棟が竣工したばかりで計算機の設定が済んでいないとのことなので、残念ながら計算機での実習はなしです。(紙の上での演習を少々やって戴きます。) 計算機での実習については、来週説明しますので欠席しないようお願いいたします。

皆様からのフィードバックを頂くため、各回とも実習テーマから適宜選んで提出をお願いするつもりです。提出物を綴じるため、ホチキスを各自持って来る(か、または借りる友人を確保する)ようお願いいたします。あと、レポート用紙も必要ですが、実習が進めばプリンタ出力の裏に書けばいいようになると思います。紙はすべてA4版を使用すること。

課題は内容によって時間内提出のものと、次回までに提出のものを適宜(必要なら両方)出します。これらはすべて集計して、出席点とレポート点を差し上げます。最終成績はこれらによってつけ、試験は行なわないようにしたいと考えています。(私は学生時代試験が大嫌いでした。) その分、日頃の誠意(まめに出席して課題をこなすこと)が重視されると思ってください。

当面教科書は指定せず、資料を毎回配布します。この辺は後日また変わるかも知れませんが、参考書として「情報処理入門補遺/UNIX 対応版」を指定しますので、生協で入手してください。では、これから半年よろしく申し上げます。

*筑波大学大学院経営システム科学専攻

1 計算機と情報

まず、この節の表題どおり、「計算機というのは何をする装置か?」という質問を投げかけてみたい。あなたならどのような答を考えつかれるか? たとえば次のようなものか?

△ 計算機とは、計算をするための装置である。

それだったら、計算機と電卓は同じものか? そうではないでしょう?

ここで「大規模な」とか「大量の」とか「高速に」とつけ加えようとする方は、残念ながら「いまいち」である。確かに世界最初の電子計算機は数表のようなものを大量に「計算」するために作られたのだけれど、現在の計算機の用途としては「計算」はごくマイナーな用途である。たとえば JR の切符を予約したり、建築図面を描いたり、英文抄録をまがりなりにも和訳したりすることはあまり「計算」らしくない。そこであなたは次のような答を考えつかれるかも知れない。

○ 計算機とは、情報を処理するための装置である。

まあ、確かにそうなのだが。しかし、この定義ではあまりにもぼくぜんとしていませんか。「情報」とは何か? 「処理する」とはどうすることか?

そこで、まず「情報」をもっと具体的な、目に見える形で表すことを考える。つまり、世の中にはさまざまな情報が満ちあふれているが、その最小単位は「はい」「いいえ」、「ある」「ない」、のように2つのうちどちらである、という情報だと考えることができそうである。これをもっとコンパクトに書くため「1」「0」の文字を使うことにしよう。この情報の最小単位のことを「ビット(bit)」と呼ぶ。

世の中の(少なくとも計算機で扱えるような)情報はすべて、この最小単位であるビットを複数個組み合わせることで現すことができる。たとえば交通信号の色(赤、黄、青)を表すには次のようにビットを2つ組み合わせればよい。

ビット0	ビット1	色
0	0	赤
0	1	黄
1	0	青
1	1	使用せず

つまり、2ビットを使用すれば $2 \times 2 = 4$ 通りの場合が表せるから、そのうち3つを3色に対応させ、あとの1つは使わなければよい。次に、十字路口には信号が2組(東西方向と南北方向というように)設置されているから、その組み合わせを考えると次のようになるだろう。

ビット0	ビット1	ビット2	ビット3	東西方向	南北方向
0	0	0	0	赤	赤
0	1	0	0	黄	赤
1	0	0	0	青	赤
0	0	0	1	赤	黄
0	0	0	1	赤	青

今度は「使用せず」のところは略した。交通信号は両方同時に青にはならないから、4ビットで可能な $2^4 = 16$ 通りのうちわずから5つしか使っていない(もったいないから3ビットで済ます、とい

う話はとりあえずおいておく)。このように、ビット列を連結することで場合の組み合わせを簡単に扱うことができる。

一般に、長さ n のビット列は、 2^n 通りの場合分けを表すことができる (わかりますね?) 例えば、 $2^{16} = 65536$ なので、16 ビットあればあらゆる漢字を含んだ日本語の文字が区別できる (し、実際そうしている)。¹

また、ビット列は集合のようなものを表すのにも便利である。例えばあなたの物理実験のチームが 8 人いたとして、0 ビット目～7 ビット目をそれぞれの人に対応させるものとする。ここで、「今週出席した人の集合」という情報は、出席した人を 1、欠席した人を 0 に対応させるとして、例えば

01011101

のように長さ 8 のビット列で表せる。

もちろん、数もビット列で表せる。それには、普段使っている記法 (十進法) と同様に位どり記法を使う。たとえば

107

と書いた場合、これは一番右が「一の位」、次が「十の位」、一番左が「百の位」だから、合わせて百と七を表す。なお、十進法だと各位の数字は 0～9 のどれかで、桁が進むごとに表す値は十倍ずつになるわけだ。²

これが、ビット列だと各位の「数字」は 0 と 1 しかなく、そこで各桁ごとに表す値は 2 倍ずつになる。つまり

01011101

だと右から順に 1 の位、2 の位、4 の位、8 の位、16 の位、…となるから、このビット列は (2 進表記の正整数として見た場合) $64 + 16 + 8 + 4 + 1 = 93$ を表すことになる。一般に、 n ビットのビット列で $0 \sim 2^n - 1$ の数が表せるわけですね。

ということで、以下では「情報」を「ビット列」で現せるものとする。そうすると、計算機にはビット列を与えることになる。そして、計算機は内部でそれを「処理」した後、結果の「情報」をやはりビット列の形で返してくれるようにできるだろう。「処理する」という言葉はあいまいな感じがするので、もっと直接的に「加工する」といい直して、以下では

◎ 計算機とは、ビット列を加工するための装置である。

ということにする。たとえば、10001 というビット列と 00011 というビット列を与えて「足し算」という加工をしてもらおうと 10100 というビット列が得られるわけである。

ここで注意しておきたいのは、それぞれのビット列が何を意味しているか、という「解釈」のしかたはあくまで人間にまかされているということである。つまり、先に出てきた「01011101」というビット列が実験の出席仲間を表しているのか、数 93 を表しているのか、93 だとしてそれが BF や GF の数なのか、試験の点数なのかはあなたが決めることである。

そして、計算機は与えられたビット列を決まったやり方で加工して結果のビット列を求めるといふ点では高い能力を持つが、そのビット列をあなたが解決したいと思う問題にあわせて解釈した時実際に正しい解になっているかどうかはそれとはまた別のことがらでありむしろそちらの方が現実の問題を解く上ではむずかしい課題である。

問題 1-1 次の情報をビット列で表すやり方を考案してみよ。

¹なお、計算機の世界では慣例として数を 0 から数えはじめることがとても多いので、そのつもりでいてください。

²当たり前ですか? うちの娘は小学校 1 年生なので… 子供に位どりの概念を教えるのって大変ですよ。

- a. 来月のすべての日のうち、どの日に学校に登校するかの情報。
- b. 先月のすべての日について、「良かった日」「悪かった日」「普通だった日」はそれぞれどの日だったかの情報。
- c. あなたの手持ちの衣服のうち、現在どれを着用しているかの情報(下着の情報は略す)。

問題 1-2 次の数(2進表記)を十進表記(要するに普通の書き方)に直せ。(どうやったかも簡単に示す。)

- a. 000110011
- b. 111111111
- c. 111111100

問題 1-3 次の数(十進表記)を2進表記に直せ。(やり方は自分で工夫する。どうやったかも簡単に示す。)

- a. 23
- b. 123
- c. 1023

問題 1-4 次の足し算(と引き算)をせよ。すべて2進表記のこと。

00110110	00110110	00110110	11001100
+) 01000001	+) 00110110	+) 00111111	-) 00111001
-----	-----	-----	-----

問題 1-5 4ビットで正の整数を表すと0~15まで表せる。その対応表を書け。また、負の数まで含めて表すとしたら、どういうやり方で表すのがいいと思うか。できれば2通り以上考えよ。(注: 4ビットでは16通りしか表せないから、負の数を含めるとだいたい-8~+8くらいの範囲の数だけが表せることになる。)

2 計算機の見ためと構造

計算機のことをより正確には「電子計算機」(electronic computers)と言うように、計算機の内部では電子回路によって上述のビット列の情報を扱う。具体的には、「電流が流れている/いない」「電圧が掛かっている/いない」の二者択一によってビットの0/1を表しているわけである。³計算機の内部で、この電子回路を駆使して情報(ビット列)の加工を行なう部分のことをCPU(Central Processing Unit、中央処理装置)と呼ぶ。

ところで、ビット列を加工するに際してはそれを一時的に保管ないし記憶しておくための場所が必要である。(たとえば、沢山の数を合計しようとしたら、その中間結果をメモしておく紙が欲しいでしょう?) そのための部分のことを、主記憶装置(main memory、略してメモリ)と呼ぶ(図1)。メモリもやはり電子回路できていて、電流や電圧の有無を記憶しておくことができる。その容量はビット数で計るが、普通は8ビットを1バイト、1,024バイトを1KB(キロバイト)、1,024KBを1MB(メガバイト)として最近ではメガバイト単位で表すことが多い。

CPUとメモリは計算機システムの中核部分をなす。CPUはメモリから加工すべきビット列を取り出し、加工し、結果を再びメモリに格納する。CPUの内部にはあまり多くの情報は蓄えられ

³これは実は話が逆で、本当は電子回路ではこのような二者択一の情報が一番扱いやすいから、それで電子計算機はビット列を扱うように設計されているのである。

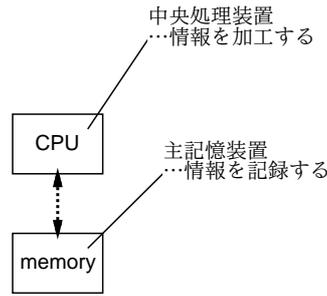


図 1: CPU とメモリ

ないのだが、この取り出し＝加工＝格納は非常に高速に行なえるので、結果としてメモリに蓄えられた大量の情報を短時間で処理していくことができる。もちろん、これらの情報(ビット列)はすべて(電子回路だから)電流や電圧の有無で表されている。

しかしもちろん、電流や電圧の有無は直接目でみることはできない。それではせっかく情報を加工してもらっても人間がその情報を受け取ることができない。逆に、計算機に加工してもらうための情報があったとして、それを計算機に伝えることも必要である。このような、人間との間で情報をやりとりする装置のことを入出力装置(Input/Output Unit、I/O などとも書く)と呼ぶ。なお、人間→計算機の伝達を入力、計算機→人間の伝達を出力と呼ぶとに(伝統的に)なっている(図 refc0-io)。

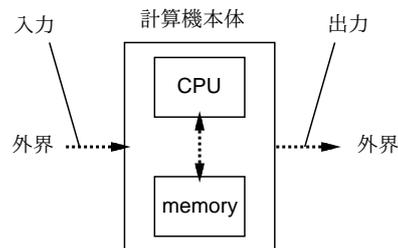


図 2: 入力と出力

たとえばごく素朴なやり方として、電流(か電圧)の有無はスイッチを使えば人間が制御できるし、豆電球や発光ダイオードをつなげば見ることができる。しかし、山のようにスイッチが並んでいたり沢山の点滅するランプが見えてもあまり嬉しくはないですね?⁴

現在では、入出力装置としてはもっと人間に扱いやすいものが使われる。そう、あなたの目の前にあるキーボードやマウスは入力装置であり、画面は出力装置である。

問題 2-1 ここで説明したもののほかに、計算機の入出力装置としてどんなものがあると思うか? 実際に知っているものでも、今ここで考え付いたものでもいいから 3 個以上記せ。

問題 2-2 計算機内部の分解ビデオを見て、どう思いましたか。簡単な感想を書け。⁵

⁴しかし少し古い映画などでは計算機というとやたら沢山のスイッチが並んでいて、ランプが点滅していたりしますよね。

⁵これは大演習室でないと上映できないのでこの次にします。

3 計算機との対話

さて、利用者が(たとえばあなたが)計算機を使うというのはどういうことか、もう少し考えて見る。計算機は情報を加工する装置だから、あなたはまず入力装置(たぶんキーボード)から計算機に情報を投入する。そうすると、計算機はそれを加工してくれ、結果の情報を出力装置(たぶん画面)に提示する。あなたはそれを見て、満足すれば次の仕事に進んでもいいし、もう少し変えた情報を加工して見たければその情報を投入することになる。このように、計算機と利用者は「利用者→計算機→利用者→計算機→…」のように交互にイニシアチブを取りながら進んで行くものと考えられる(図3)。

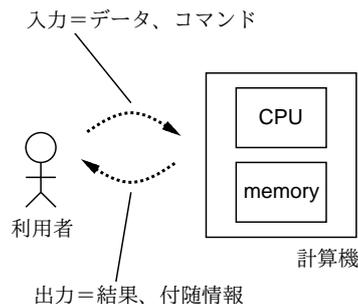


図 3: 計算機との対話

ところで、上で(情報を)「加工して…」と記したが、具体的にはどのように「加工」するのだろうか? 例えば、小遣い帳の計算をしたいのなら使った金額(という数値情報)の合計をとる、という加工をしたいだろうし、その他場合によるわけだ。すると、「どのように加工するか」という「情報」も計算機に与えなければならない。逆にいえば、様々な加工のしかたを自由に選べるという柔軟性があるからこそ、計算機はここまで広く使われるようになってきているわけだ。

なお、計算機の世界では一般に、加工される情報のことを「データ」、どのように加工するかを示す情報のことを「指令」ないし「コマンド」とよぶ。皆様は特に計算機で処理したいデータが沢山ある、というよりはどのように計算機を扱うかを学びたいわけなので、データよりはコマンドをたくさん打ち込むはめになる…ものと思うので、そのつもりで。

問題 3-1 銀行 ATM(ATM は立派な計算機の手先である)での貯金引出しを例にとって、利用者(あなた)から計算機への入力と、計算機からの出力(表示やお金や音声)をすべて順番に列挙し、上述の計算機=利用者サイクルを何回回っているか数えよ。また、どの入力が「データ」でどの入力が「コマンド」かも示せ。

問題 3-2 その他のシステムないし機能を1つ選んで同様にやってみよ。例えば送金振り替えではどうか? JR 指定券自動発売機ではどうか?

4 プログラムと手順

ここまででは、ある加工に対して、それを行うためのコマンドが既に存在していて、それを示す情報を計算機に与えればいい、という前提で話をしてきた。しかし、現実には自分のやりたいことすべてが予めコマンドとして用意されている、などということはもちろん不可能である。

ではどうするか? それには、「どのような加工をしたいか」という情報を、(またまたビット列として)計算機に与えればよい。この情報のことを(散々聞いたことがあると思うが)「プログラム」という。実は「既にあるコマンド」というのは、「あらかじめ計算機に入力されて蓄えられているプログラム」に他ならない。

言い替えれば、計算機というのは実はプログラムに従って情報を処理 (= ビット列を加工) する装置なわけである。そして、どのような処理であっても (たとえその計算機が製造された時には創造だにされなかったようなものでも)、その処理方法をプログラムとして与えさえすれば処理できるようになる、というところに計算機の特徴があるわけである。というわけで、この科目の目標の1つとして皆様にプログラミングとアルゴリズムについて学んでいただくことになっているわけである。

なお、ここで「プログラム」と「アルゴリズム」ないし「手順」の違いを整理しておく。「アルゴリズム」ないし「手順」といった場合、それは加工の方法自体をいう。例えば駒場地区正門から渋谷駅まで歩いて行く行き方、といったものである。一方、「プログラム」といった場合はアルゴリズムを計算機に与えられるような形に表現したものをいう。例えば道筋を他人に教えるためには、行き方を記した地図やメモといった具体的な形に表現する必要がある。

問題 4-1 「アルゴリズム」と「プログラム」、「行き方」と「行き方を示した地図」のような関係にあるものの例を他にも挙げてみよう。

問題 4-2 「行き方」を具体的に表現する方法として、どんなものがあるか3つ以上挙げよ。それらの方法で、自分の住所からもより駅 (ないしバス停) までの行き方を表現してみて、その優劣を比較せよ。

5 計算機のためのアルゴリズム

計算機で使う「アルゴリズム」ないし「手順」は、何かを求めるための具体的な計算 (ないし情報の加工) 方法でなければならない。たとえば、海外のニュースなどを見ていると気温が華氏で表示されているので、それは摂氏では何度かな、と知りたくなる。それを求めるには、華氏の温度を f として、

$$c = \frac{5}{9}(f - 32)$$

により値 c を求めればよい。これも立派なアルゴリズムである。

ところで、何をもって「具体的」というかは実は簡単ではない。たとえば、 n 個のボールを (すき間があってもいいから) 正方形の箱に平らに入れたければ、その箱の1辺の長さはボールの直径の

$$l = \lfloor \sqrt{n} \rfloor$$

倍であればいい。ただしこれは、「切り上げ」とか「平方根」とかの計算手順が具体的にわかっていれば、である。もし加減乗除だけしか使えないのなら、例えば

$$l^2 \geq n$$

なる最初の l が見つかるまで $l = 0, 1, 2, 3, 4, \dots$ を順に試していく、という手順を使うことになる。

実は、四則演算も「具体的」かどうかは議論があつてよい。たとえば小学生はどうやって四則演算をやるかの手順を一生懸命憶えさせられるわけである。しかし幸いなことに、計算機には四則演算のための機能がもともと備わっているから、それをお願いすることにして、まずは四則演算は「具体的」だということにする。

問題 5-1 仮に、かけ算と割り算は「具体的でない」(つまり使えない)ものとする。足し算と引き算と数の大小比較は使ってよいとして、次のことをするアルゴリズムを考えよ。ただし x 、 y は正の整数とする。

- a. 数 x が奇数か偶数かを判定する。
- b. 数 x と y の積を求める。
- c. 数 x と y の最大公約数を求める。

6 アルゴリズムの表記方法と PAD

アルゴリズムを計算機での処理に使おうと思うと、「具体的」の他にも色々問題がある。まず、上の温度の例では「華氏の温度を f として」などと書いてあったが、人間が手で計算するならそれでいいとしても、計算機で処理する場合にはそこで「華氏の温度を計算機に読み込ませる (入力する)」という動作が必要である。その次に式にしたがって四則演算を行ない、最後に結果を「人間に見えるように表示する (出力する)」動作が必要になる。これらのことを順番におこなって始めて、計算機で仕事がこなせるのである。

そこで、計算機の世界ではもっとこの「入力」「出力」「順番の動作」がはっきりするような書き方を工夫することが古くから行なわれている。その古典かつ代表が「フローチャート (流れ図)」だが、現在ではこれは様々な弱点があつてよくないとされている。ここでは PAD 図と呼ばれるものを使用する。上の温度変換の手順を PAD で記したものを図 4 に示す。

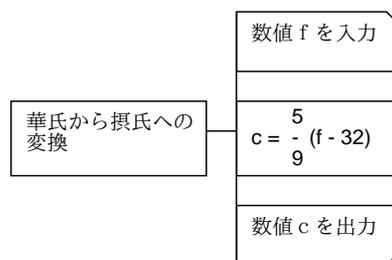


図 4: 華氏摂氏変換の PAD

ここで、箱は何らかの計算機による処理を表している。特にそれが入力、出力の場合には、それぞれ右上、右下が斜めに欠けた箱を使う。箱の連なりが縦線で結んである場合には、それらを上から順にやっていくことを示す。そして、ある箱から右に引き出して記述がある場合には、その箱に書いてあるのは概略で、より具体的には右側の記述によること (詳細化) を示す。

ここにあるように、どんなプログラムでもまずそのプログラムが全体として何をするかを書き、その詳細化として手順を書いていくのがよい習慣である。

練習 6-1 次のような計算の手順を PAD で書いてみよう。今回は四則演算は使ってよい。

- a. 二つの数 a 、 b を入力し、その合計を出力する。
- b. あるものの長さが X メートル Y センチ Z ミリの形で表されていたとして、それをセンチ単位に換算する。(例えば 1 メートル 10 センチ 5 ミリ \rightarrow 110.5 のように。)
- c. 同じ日の 2 つの時刻 A 時 B 分 C 秒と X 時 Y 分 Z 秒を読み込み、その間の時間を秒単位で出力する。

7 プログラミング言語 Pascal

「プログラム」とは、アルゴリズムを実際に計算機与えられる形で表現したものだった。さっきやったように、アルゴリズムのようなものを書き表す方法はいろいろある。これまでは日本語

とか数式とか PAD 図を使って書き表していた。それに対して、計算機で計算させるためには日本語や数式ではやりにくいし、PAD 図では読み込ませにくいから、もっと杓子定規な書き方の規則を設けて、それに従って書く。これがプログラムである。

その「書き方の規則」も実は様々な流儀がある。つまり各種のプログラミング言語に相当する。どの流儀も、それなりの長所と弱点がある。ここでは Pascal と呼ばれる言語を用いる。Pascal はチューリヒ工科大学の Wirth 教授が設計した言語であり、教育用によく使われている。また、強い型検査機能のためプログラムの間違いが発見しやすい。反面、大きなプログラムや、システムプログラミング (各種システム機能を駆使するようなもの) にはあまり向いていないとされている。では、先の華氏と摂氏の変換プログラムを Pascal で記述してみる。

```
program sam1(input, output);
var c, f: real;
begin
  write('degree F> ');
  readln(f);
  c := (5.0 * (f - 32.0)) / 9.0;
  writeln('degree C = ', c:5:2)
end.
```

先の PAD と比べてみると、いろいろ細かい記述が増えているなあと思う。

1. プログラムは「program プログラム名 (input, output);」という書き出しで始まらなければならない。
2. 値を表す名前 (変数) は「var 変数名, ...: real;」によって予め使用を予告しておかななければならない。
3. begin から end. までの間に手順を順次; で区切って書く。なお、この区切られた各 1 手順を Pascal では「文」と呼ぶ。
4. 入力は readln、出力は writeln で表す。なお write は改行なしになること以外は writeln と同じ。
5. 加減乗除は+*/でそれぞれ表す。1 行に書かないといけないので、適宜 () でくくる必要がある。
6. 変数に値を設定するには代入 (「:=」と呼ばれる記号を使う。代入は「等しい」とは違う意味 (cf. $x := x + 1.0$)。

さらに writeln では文字列を書く時には'' で囲む、数値を書く時には「値:幅:桁数」の指定を行なうと、全体として指定した幅以上の文字数を用いて、少数点以下を指定した桁数ぶん表示する、といったこともおぼえる必要がある (ああ面倒だ!) ⁶

要は、プログラミング言語というのは計算機に対して実際にアルゴリズムを実行する際のありとあらゆる細かい所まで指示できるように決めた形式であり、だからプログラムのどこか少しでも変更すると計算機の動作もそれに相応して変わるか、(もっとよくあることだが) そういう風には変えられないよ、と怒られることになっている。いくら怒られても偉いのは人間であって計算機ではないのだから、そういうものだと思って許してやって頂きたい。

実際にこれを計算機で動かすための操作は来週やるが、とりあえず動く時の様子だけでも見ていただく。

⁶桁数を指定しない場合には、指数形式での表示になる。

```
% pc sam1.p
% a.out          ←プログラムの実行開始
degree F> 60     ←データの入力
degree C = 15.56 ←結果の出力
%
```

苦勞のわりにはあんまり大したことはない感じだが、まあ初心者第1歩ということで、そうがっかりしないで戴きたい。この先は来週のお楽しみということで。

練習 7-1 練習 6-1 で描いた各 PAD を Pascal のプログラムに変換せよ。

A 本日の課題

今日は第1回ですので(計算機も使えないことだし)、次回までにやっていただく課題はありません。

ただ、今日時間中にやっていただいたものをそのまま綴じて提出していただきます。量が多いので、全部やってある必要はありません。

あと、綴じた裏面に今日の日付け、学籍番号、氏名を記入していただき、さらにフィードバックをいただくため以下のアンケートの答えを記入してください。そんなに大量に書く必要はありません。

- Q1. これまでに計算機と縁がありましたか? あつたとすれば、どんな経験がありますか? なかつたとすれば、計算機をどんなものだと思っていましたか?
- Q2. 本日の内容のうち、知っていた事柄はどれくらいありましたか? また、興味深く思った事があれば挙げてください。
- Q3. 本日の全体的な感想と今後の要望をお書きください。