

「情報処理」1年文I/IIクラス9-10 #6

久野 靖*

1994.11.28

0 本日の目標

さて、駒場祭の疲れも取れたと思うので、また通常ペースで進みたいと思います。本日の目標は次の通り。とくに WWW のページを書くとき世界中に情報を発信できますから、頑張りましょう。

- if を使ったプログラムの復習。
- 新たに反復 (繰り返し) を学ぶ。
- WWW のページを書けるようになる。

1 前回課題の復習

さて、**4B** の課題 7 の PAD 図を図 1 に再掲し、対応する Pascal プログラムを一応示しておこう。

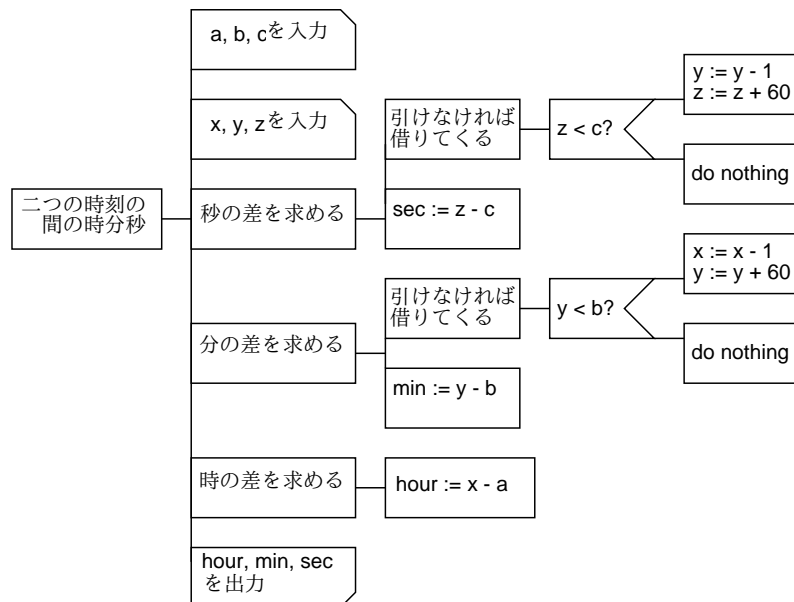


図 1: **4B** の課題 7 の PAD(再掲)

*筑波大学大学院経営システム科学専攻

```

program sam7a(input, output);
var a, b, c, x, y, z, hour, min, sec: integer;
begin
  write('a = '); readln(a);
  write('b = '); readln(b);
  write('c = '); readln(c);
  write('x = '); readln(x);
  write('y = '); readln(y);
  write('z = '); readln(z);
  if z < c then begin y := y - 1; z := z + 60 end;
  sec := z - c;
  if y < b then begin x := x - 1; y := y + 60 end;
  min := y - b;
  hour := x - a;
  writeln(hour:1, 'jikan ', min:1, 'fun ', sec:1, 'byou')
end.

```

ところで、writelnでの表示に「時間」とか「分」といったふうに漢字を使う事は何ら問題ない。失敗している人は、「'」など文字列の外側の文字まで漢字にしてしまったためと思われる。記号類はすべて8bit(漢字の半分の幅で表示される)でなければならないことに注意!

ついでに、復習を兼ねてこれを「箱で囲む」と図2のようになる。確かに「;」の位置は「文と文のさかい目」ですね? なお、この例のように、begin-endのあるプログラムをどういう風に配置

```

program sam7a(input, output);
var a, b, c, x, y, z, hour, min, sec: integer;
begin
  write('a = '); readln(a);
  write('b = '); readln(b);
  write('c = '); readln(c);
  write('x = '); readln(x);
  write('y = '); readln(y);
  write('z = '); readln(z);
  if z := c then begin y := y + 1; z := z + 60 end ;
  sec := z - c ;
  if y := b then begin x := x + 1; y := y + 60 end ;
  min := y - b ;
  hour := x - a ;
  writeln(hour:1, 'jikan ', min:1, 'fun ', sec:1, 'byou')
end.

```

図 2: 図 1 のプログラムの「文」を箱で囲んだもの

するかは結構趣味の分かれるところで、統一解は難しい。センスを働かせて欲しい。

次は演習 8 だが、a は簡単だから飛ばしてまず b をやろう。PAD は図 3 の通り。わかりますか? まず a と b が正しい順になるように、もし逆順だったら交換する。次に、b と c が正しい順になるように、もし逆順だったら交換する。その結果 a と b の関係が正しくなくなる可能性があるので、再度 a と b を比較して、逆順だったら交換する。いいでしょう? プログラムにすると次の通り。

```

program sam7b(input, output);
var a, b, c, x: integer;
begin

```

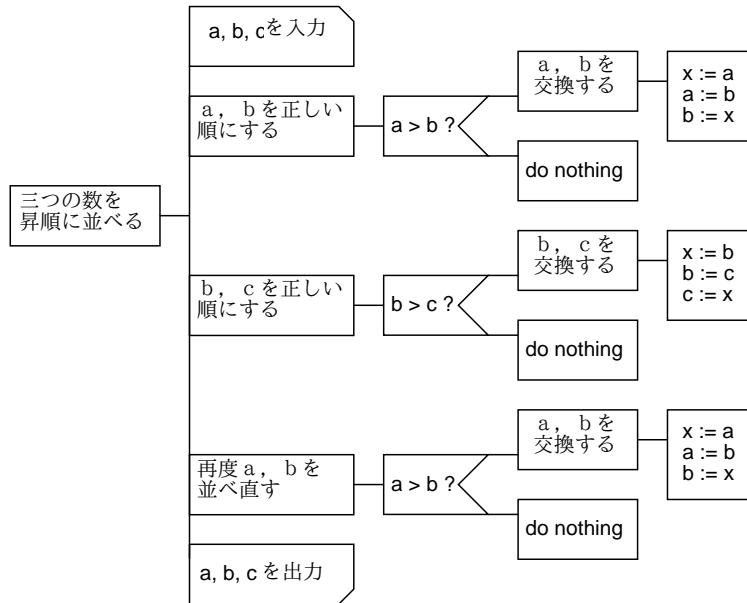


図 3: 4B の課題 8b の PAD

```

write('a = '); readln(a);
write('b = '); readln(b);
write('c = '); readln(c);
if a > b then begin x := a; a := b; b := x end;
if b > c then begin x := b; b := c; c := x end;
if a > b then begin x := a; a := b; b := x end;
writeln('a = ', a:1, ' b = ', b:1, ' c = ', c:1)
end.

```

実は、前回の資料と演習はこれのヒントになっていたのだった。お分かりかな? 「交換」という操作はこれからもよく使うと思うので、覚えておくとよい。

次は演習 8 の b だが、これは if-then-else if の連鎖が上から順に条件を調べていくことを利用するとすっきりできる。PAD 図は図 4 に示す。ところで、「year - 1868 + 1」は「year - 1867」

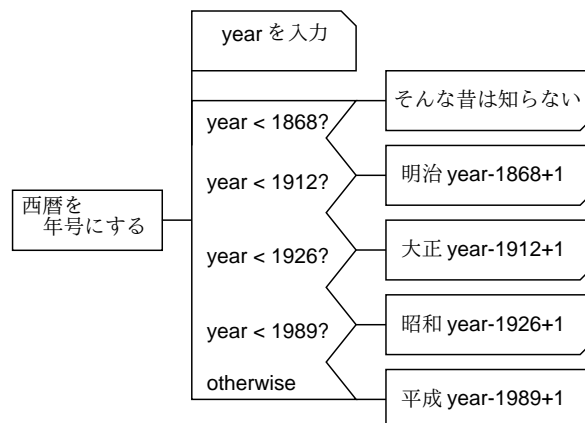


図 4: 4B の課題 8c の PAD

でいいと思いませんか? 別に間違いではないのだが、条件判定に使う数値と引き算に使う数値を

一致させておいた方が「私は気持ちがいい」からそうやっている。なお、このために計算時間が損ということは全然ない。ではプログラムを示す。

```

program sam7c(input, output);
var year: integer;
begin
  write('year = '); readln(year);
  if year < 1868 then
    writeln('too old, I do not know.')
  else if year < 1912 then
    writeln('meiji', year-1868+1:1)
  else if year < 1926 then
    writeln('taishou', year-1912+1:1)
  else if year < 1989 then
    writeln('shouwa', year-1926+1:1)
  else
    writeln('heisei', year-1989+1:1)
end.

```

ところで、「年がどこからどこまでの間」という条件を使いたいと思った人もいるでしょうね？ その説明をしないですいません。それには、

```
if (year < 1989) and (year >= 1926) then ...
```

のように、条件を「and」という演算子で結合すれば書ける。同様に、「～または～」を表す「or」という演算子と、「～でない」を表す「not」という演算子も用意されている。ただ、この問題に限っては複合条件を使うよりも if-then-else if の連鎖の性質を使う方がスマートだと思ったので、つい説明しそこなったわけなのでした。

演習 8 の d もやっておこう。これも if-then-else if です。なお、予期しない (おかしい) 入力に

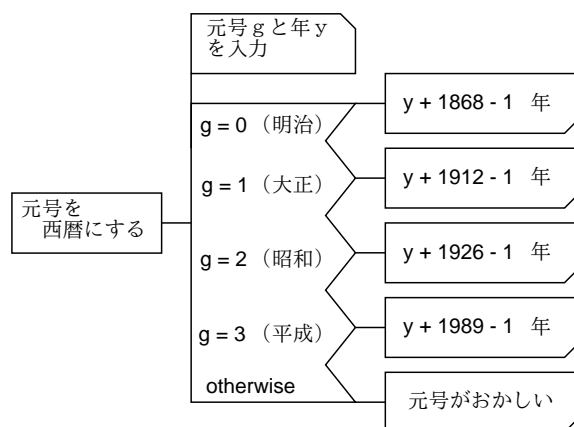


図 5: 4B の課題 8d の PAD

ついてもなるべくちゃんと「おかしい」と教えてあげるのが親切というか、常識。(自分が使う人だったらどうか考えるように。) で、プログラムはもはや簡単ですね？

```

program sam7d(input, output);
var g, y: integer;

```

```

begin
  write('gengou (0=Meiji, 1=Taishou, 2=Shouwa, 3=Heisei) : '); readln(g);
  write('y = '); readln(y);
  if g = 0 then
    writeln(y + 1868 - 1:1)
  else if g = 1 then
    writeln(y + 1912 - 1:1)
  else if g = 2 then
    writeln(y + 1926 - 1:1)
  else if g = 3 then
    writeln(y + 1989 - 1:1)
  else
    writeln('g should be 0, 1, 2, or 3.')
  end.
end.

```

2 繰り返し

さて、ここまで説明してきた PAD や Pascal の機能では、上から順番に実行していった (枝分かれはあっても) 下まで来たらおしまい、だから大したことはできない。そこでいよいよ、繰り返し (ループとも呼ぶ) について学ぼう。まず一番基本である、while 型の繰り返しを表す PAD を見て頂く (図 6)。たて線の入った箱は「繰り返しの箱」であり、その右側にくっついている部分 (ルー

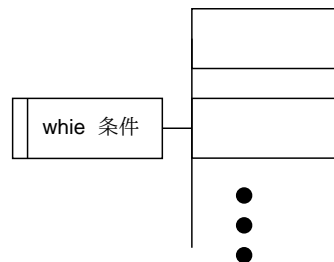


図 6: 繰り返しを表す PAD

プ本体とも呼ぶ) は繰り返し何回も実行される。ここがこれまでの 1 回しか実行されない PAD とは大幅に違っている。

では、どれくらいの回数実行されるのか? while 型の繰り返しでは、繰り返しは「条件」が成り立つ限り何回でも実行される。だから、ループ本体には必ず条件を変化させていく部分が含まれていなければならない。具体的に見てみよう。例題として、ずっと前に考えて頂いた、資料 # 2 の課題 3 を使う。まず次のものから (数はすべて正の整数)。

- a. かけ算と割り算を使わず、数 x が奇数か偶数かを求める。

ここで、次の事実を使う。

- 0 は偶数、1 は奇数である。
- ある数から 2 を引いても奇数/偶数の別は変化しない。
- 1 より大きい正の整数から 2 を引いても、0 より小さくはならない。

これらに基づいて、図7を考えた。確かに、ループ本体「 $x := x - 2$ 」は条件「 $x > 1$ 」を変化させるような操作になっている。より具体的に考えると、 x がだんだん減っていくのだから、いつかは条件が成り立たなくなってループが終わるはずである。また、ループ本体を何回実行しても、 x の奇数/偶数の別は変化しない。そして最後に、ループが終わった時は x は1以下であり、また0より小さくはないはずだから、結局0か1かのどちらかである。というわけで、0か1かを調べれば偶数か奇数か分かることになるでしょう？

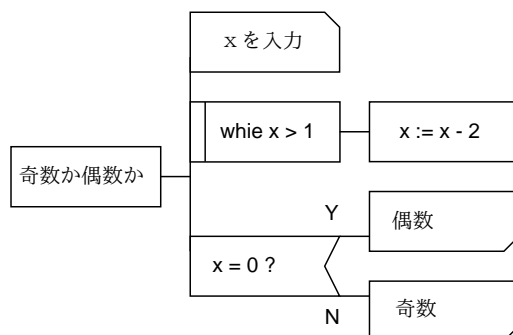
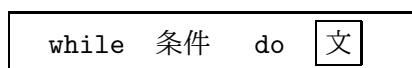


図 7: 奇数偶数判定の PAD

さて、このようなループを含む PAD を Pascal に直すのには、次の while 文を使う。



while 文はこれまで見て来た「文」が置ける場所ならどこにでも置ける。また構文(箱の形)的には if より簡単、というか else 部のない if のようなものである。では上の PAD をプログラムに直す。

```

program sam8a(input, output);
var x: integer;
begin
  write('x = '); readln(x);
  while x > 1 do x := x - 2;
  if x = 0 then writeln('guusuu') else writeln('kisuu')
end.

```

では次の問題へ行こう。

- b. 正の正数 x と y を入力し、その積を求めよ (かけ算演算は使わずに)。

これを PAD にしたものを図8に示す。ここではまずループに入る前に n を 0 にしている。そして、ループの中では n を y 増やし、 x を 1 減らしている。これはどういう意味があるだろう？

実は、このプログラムでは求める答を M とすると、常に $M = n + xy$ が成り立っている。なぜなら、最初に n を 0 にするから、確かに M は積 xy に等しい。その後、ループの中では n を y 増やす代わりに x は 1 減らすのだから、 $(n + y) + (x - 1)y = n + xy = M$ で確かにこの状態は変化しない。次に、ループの中で x を減らしていくから、確かに条件「 $x > 0$ 」に影響を与え、なおかついつかは x が 0 になってループは終わる。終わった時は x が 0 なのだから、 $M = n$ が成り立っている。つまり n に答が求まっている。

なんて面倒くさい! と思いませんか? そんな理屈を考えなくても、単に y を x 回足していることは見ればわかるじゃないか! 確かにこれくらいの例題ならそうなのだが、これから次第に複雑なプログラムになった時、それを正しく作るコツは、次の事柄をきちんと確認することである。

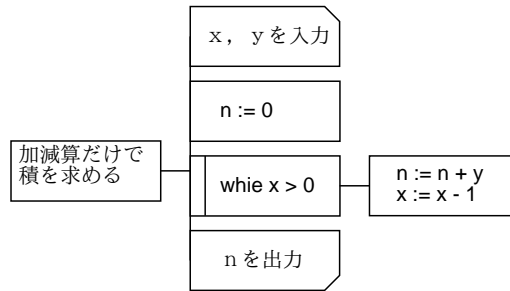


図 8: 加減算だけで積を求める PAD

- ループを通じて変化しない条件は何か?
- ループ本体に、条件の成否を変化させる部分があるか?
- そして、その変化は最後にはループが終わるような変化か?
- ループが終わった時、欲しい値が確かに求まっているか?

演習 1 図 8 の PAD を Pascal に直して動かせ。

では次の問題へ進もう。

- c. 正の整数 x 、 y を読み込み、その最大公約数を求めよ。

今度は考え方をまず説明するので、PAD からやって頂く。まず、 x と y が等しければ、最大公約数は x (そして y) そのものですね? 次に、もし x と y が等しくなければ、そのうちより小さい方を x' 、大きい方を y' とする (例によって、逆順なら交換すればいいですね)。そして、 $y'' = y' - x'$ とすると、 x と y の最大公約数は x' と y'' の最大公約数と同じである。(なぜか? x も y も最大公約数 g の整数倍なのだから、その大きい方から小さい方を引いた数も、やっぱり g の整数倍に決まっているでしょう?) 従ってこれを新たな x 、 y と考えてよい。これを繰り返していくと、 x も y も段々小さくなっていくが、 g より小さくなることはできないから (なぜ?)、最後はともに g に等しくならざるを得ない (おわかりかな?)。

演習 2 最大公約数の問題を PAD にしてみよ。さらに、Pascal にして動かせ。

なお、Pascal では「 x と y が等しくない」という条件は「 $x \neq y$ 」のように書く (単に \neq という字がなかったから)。では、ループに慣れるための練習問題もいくつか挙げておこう。

演習 3 次のプログラムの PAD 図を書け。さらに、Pascal にして動かせ。

- 正の整数 x と m を読み込み、 x を m で割った商と余りを求めよ。ただし加減算のみを使うこと。
- 正の整数 n を読み込み、その階乗 ($n \times (n - 1) \times \dots \times 1$) を求めよ。
- 正の整数 n を読み込み、 $x^2 \geq n$ を満たすできるだけ小さい正の整数 x を求めよ。

3 WWW の情報を作る

3.1 ネットワークとアドレス

これまで、WWW の情報は見る一方だったが、そろそろ自分から情報を発信してみたいと思いませんか? (電子メールでも情報は発信できるが、あくまでも個人宛だった。) そのためには、WWW の「仕組み」をもう少し知っておく必要がある。

まず、WWW の情報には世界中のものが含まれていることから、その情報がどこか 1 箇所、まして駒場システムのどこかに保管されているということは考えられませんね？ 一方、例えばこの科目のホームページは当然、駒場のシステム内に入っている。

つまり、WWW というのは世界中にちらばった多数の情報格納場所の集合体として存在する。その 1 つずつの格納場所のことを「WWW サーバ」と呼ぶ。そして、我々は xmosaic を使って WWW を眺めている間に、必要に応じて世界中の WWW サーバに対し Internet(我々が接続しているネットワーク) を通じてアクセスしているわけである。

Internet では、すべての計算機(やその集合体)に対してそれを指示する名前がつけられている。例えば皆様が使っている計算機(端末サーバ)は

```
xss25.komaba.c.u-tokyo.ac.jp
```

(複数あるので番号のところは適宜変わってくる) などの名前がついている。これは右から「日本の、教育機関の、東京大学の、計算センタの、駒場地区の、xss25 というホスト」を表している。(米国などでは郵便のあて先を番地、通り、町、市、州つまり小→大の順に書くので、いわば米国式ということになる。) また、前回やったように皆様のメールアドレスは

```
ユーザ名@komaba.c.u-tokyo.ac.jp
```

であるが、これは南棟のどのホストでもメールは 1 箇所ですべてまとめて扱うから「代表番号」のようになっているわけである。実際にはメールは

```
mail.komaba.c.u-tokyo.ac.jp
```

というホストが扱っている。

3.2 WWW サーバと URL

さて、WWW の話に戻って、南棟の WWW サーバ計算機は

```
www.komaba.c.u-tokyo.ac.jp
```

という名前になっている。ところで、WWW の場合は 1 つのサーバに多数の「情報ページ」が入っているため、これを指定する必要がある。そこで例えば駒場地区のページだったら

```
http://www.komaba.c.u-tokyo.ac.jp/index.html
```

```
http://www.komaba.c.u-tokyo.ac.jp/~kuno/jouhou94.html
```

```
http://www.komaba.c.u-tokyo.ac.jp/~kuno/qa/index.html
```

```
http://www.komaba.c.u-tokyo.ac.jp/~kuno/qa/chgcolors.html
```

などのようにサーバ名とその中の情報ページを表すファイルを結合して特定のページを表す。ファイル名の部分の最初に「~ユーザ名」があるのは各ユーザが個別に持っているページ、そうでないものは WWW の管理者が管理しているページである。見て分かるように、多数のページを整理するためにディレクトリ構造を使えるようになっている。また、ディレクトリ名だけでファイル名が省略された場合はファイル名として「index.html」が使われる。つまり上の例のうち 1 番目と 3 番目は

```
http://www.komaba.c.u-tokyo.ac.jp/
```

```
http://www.komaba.c.u-tokyo.ac.jp/~kuno/qa/
```

のようにしてもよい。

なお、これらの指定の先頭にある「http:」というのは「WWW の情報ページ」を表す指定で、これ以外にも「file:」「news:」「ftp:」などいくつもの情報の形態がある。そのため、この記法を(多くの情報の形態を統合してありかを表す記法なので)URL(Uniform Resource Locator)と呼ぶ。

3.3 個人のホームページの作成

いよいよ自分の情報ページを作ってください。我々のサイトでは、個人の情報ページは各自の「WWW」という名前のサブディレクトリに置くことになっているので、まず「mkdir WWW」を実行する。その後 Mule でファイル「WWW/index.html」を作成すればそれがあなたのホームページということになる。

さて、次の問題はそのファイルの中はどんな風にかいたらいいかということで、実はこの内容は「HTML」(Hyper Text Markup Language)という「言語」に従う必要がある。この「言語」はちょっと Pascal などに似ているが、なにしろ最後は人間が見るものなのでずっと適当でよい。ご安心を。まずとりあえずおすすめの形として、次のように打って保存していただきたい。

```
<TITLE>〇〇のホームページ</TITLE>
<BODY>
<H1>〇〇のホームページです。</H1>

    こんにちは。〇〇です、よろしく。<P>
</BODY>
```

大体分かるように、HTML では「ここから…ここまでが XXX」というのを「<XXX>…</XXX>」として表す。そして、各ページはそのページを表すタイトル (xmosaic の窓の上部に表示される) と本体から成り立っている。

本体の中身は、基本的には「表題」と「段落 (パラグラフ)」から成っている。表題はその字の大きさによって「<H1>…</H1>」「<H2>…</H2>」など数種類ある。段落は、いちいち「段落はじめ」というのはさすがに面倒なので、最後に「<P>」をつけて「ここで段落終わり」とだけ示せばよい。

なぜこんな面倒をするのか? それは、Mule では全部字の大きさが同じだから、「どこは表題で大きく」というのは別途指示する必要があるし、段落も xmosaic の窓の大きさが変わると 1 行何文字かが変わるので、適宜詰め合わせ直すために「ここまでが段落」という情報を指示しておく必要があるからである。このように、内容とは別に表題や段落などの情報の指示を含めることを「マークアップ (印つけ)」と呼ぶ。HTML の「Markup」はその意味である。

演習 4 上の例のような内容をファイル WWW/index.html として作成し、xmosaic の中で「Open」ボタンを押して「http://www.komaba.c.u-tokyo.ac.jp/~自分のユーザ名/」を指定して開いてみよう。または、科目のホームページから「受講者のホームページ一覧」というページへ行き、そこで自分の学籍番号をつづいてもよい。うまく行ったら、段落を書き足したり表題の種別を H2、H3、H4 などに変えて保存し直し、xmosaic の「Reload」ボタンで再表示して変化を確認せよ。

3.4 よく使う HTML の記法

さて、あなたのページを「かっこよく」するためにもう少し HTML の勉強をしよう。まず、前回の宿題でああなたのトレードマークを作ってもらいましたね? これをページに貼りつけよう。ただし、そのためには絵を TIFF から gif と呼ばれる形式に変換し、なおかつ WWW サブディレクトリの下に置く必要がある。それには例えば次のようにする。

```
% tifttopnm TIFF ファイル名 | ppmtogif >WWW/mark.gif
```

なお、「|」はあるプログラムの出力を次のプログラムの入力に接続すること、「>」は出力を指定したファイルに書き出すことを意味している。(出力ファイルの名前は別に mark.gif でなくてもいいが、ファイル形式を表すため最後は.gif にしておく方が無難である。) この絵をページに取り込むには

```
<IMG SRC="mark.gif">
```

という指定を埋め込みたい場所に書けばよい。

演習 5 あなたのホームページの<H1>の直後に上の記述を埋め込んで、確かにページ中にイメージが現れることを確認せよ。

次は、箇条書きである。この資料にもよく出てくる、各項目の先頭に黒丸 (●) が置かれた箇条書きは例えば次のように書けばできる。

```
<OL>
<LI>自己紹介<P>
<LI>今日作ったプログラム<P>
<LI>本日のアンケートの回答<P>
</OL>
```

演習 6 あなたのホームページの中に上と同じものを挿入して、箇条書きが現れることを確認せよ。

さて、いよいよ WWW の特徴である、「クリックすると別のページへ飛ぶ」をやろう。それには、「…」を使う。ここで「…」のところに書かれた部分が下線付きで表示され、そこをクリックすると「指定」に書かれたページへ飛ぶことになる。「指定」は基本的には前述の URL 形式で書くが、ただし同じディレクトリにある別のファイルに飛ぶ時だけは単にそのファイル名のみを (ディレクトリ指定なしで) 書くことが許されている。だから上の箇条書きを

```
<OL>
<LI><A HREF="selfintro.html">自己紹介</A><P>
<LI><A HREF="sampleprog.html">今日作ったプログラム</A><P>
<LI><A HREF="computers.html">計算機についてどう思うか</A><P>
</OL>
```

のように直すと、箇条書きの各項目がつつけるようになる。

演習 7 箇条書きを上のように直して、なおかつ「cp WWW/index.html WWW/selfintro.html」などとして参照されるファイルをとりあえず最初の HTML ファイルのコピーとして用意し、つつけることを確認せよ。

演習 8 自己紹介のページをちゃんと作れ。最低限、名前と学籍番号と趣味と好きな食べ物を書くこと。項目ごとに適当な表題を使うのがよい。

演習 9 本日作った演習 1 のプログラムを表示するページの内容を作れ。なお、ただ打ち込むと段落として詰め合わさってしまうので、

```
<PRE>
program ...
...
</PRE>
```

のように<PRE>で囲むこと。プログラムだけでなく、どこを苦心したか、どう考えたかなどの内容も含める。なお、すでに別のファイルにあるプログラムを Mule で取り込むには「^Xi ファイル名 [RET]」を使用すればよい。

演習 10 「計算機についてどう思うか」のページ内容を作れ。形式や分量は自分の好きにしてよい。

A 本日の課題 **6A**

本日の課題はすべて WWW のページとして作成して頂きます。演習 1、演習 8、演習 9、演習 10 を行ってください。なお、「計算機についてどう思うか」のページの中に、「プログラミングについて」と「WWW について」を含めるようにしてください (もちろん、それ以外にも電子メール、お絵描き、Mule、日本語入力、などいくらでも含めて構いません)。それをもってアンケートにかえます (が、要望と質問があれば、「情報処理の科目に対する要望と質問」のページを作ってお下されば拝見させていただきます)。

B 次回までの宿題 **6B**

次回までの課題は演習 2(Pascal はできた人のみでよい)、および演習 3 の a~c のうちどれか 1 つ (PAD と Pascal の両方) です。また、これは特に提出ではありませんが、各自の WWW ページを充実させる (見栄えをよくする、有意義な情報を増やす、など) こともお願いします。適当な時期を見て WWW 人気投票をやりましょう。(上位の人にボーナス点をあげますので、プログラムが弱い人は挽回のチャンスですよ。) 提出するレポートの番号は **6B** です。アンケートは次の通り。

- Q1. プログラムにおける反復 (ループ) について理解しましたか? しなかったとしたら、どの辺に問題がありますか? したとしたら、どの辺が「きも」だと思いましたか?
- Q2. WWW を見るだけの時と、情報を書けるようになってからとで、見かたがどんな風になりましたか? どんな情報を提供してみたいですか?
- Q3. その他、感想、要望、質問があればどうぞ。