

「情報処理」1年文I/IIクラス11-12 #8

久野 靖*

1995.12.4

0 本日の目標

遅まきながら、駒場祭おつかれ様でした。さて、2週間前にやったループのところから再度説明して、いくらか新しい内容も追加します。そしてその後はいよいよ、WWWのページを作ってください。本日の目標は次の通り。

- ループの使い方についてももう少し詳しく学ぶ。
- うまくできたループアルゴリズムを見聞する。
- WWW ページを作って自分の自己紹介を入れる。

1 ループの復習と前回の問題の解説/☆

ループというのは「プログラムのなかで、繰り返し実行されるような部分」をいうのでしたね。たとえば加減算だけで積を求めるプログラムのPADを図1に再掲しておく。これをPascalに直すと次の通り。

```
program sam8a(input, output);
var x, y, n: integer;
begin
  write('x = '); readln(x);
  write('y = '); readln(y);
  n := 0;
  while x > 0 do begin n := n + y; x := x - 1 end;
  writeln('x * y = ', n:6)
end.
```

*筑波大学大学院経営システム科学専攻

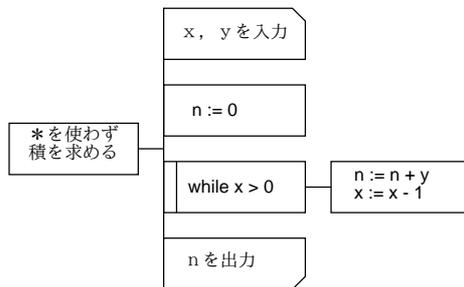


図 1: 加減算だけで積を求める

このように、変数を1ずつ減らして(または増やして)行って、ある値になったらやめる、というのはすごくよく出て来るパターンである。順序が前後するが、次の問題もそうである。

- 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。

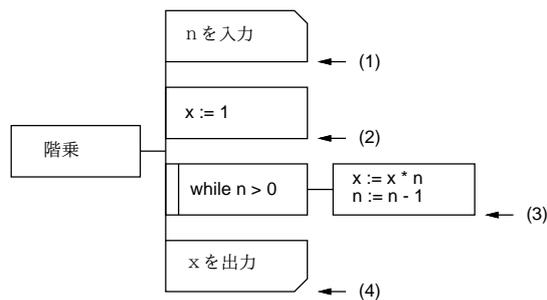


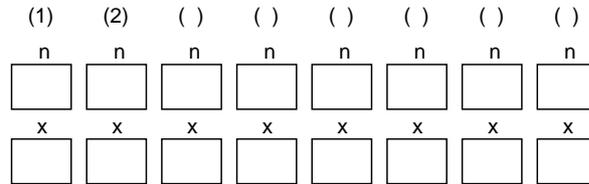
図 2: 階乗

```

program sam8b(input, output);
var n, x: integer;
begin
  write('n = '); readln(n);
  x := 1;
  while n > 0 do begin x := x * n; n := n - 1 end;
  writeln('factorial = ', x:1)
end.
  
```

なお、 x を最初 1 にしておくことと、0 を掛けてしまうと 0 になってしまうのでその手前でやめることが「コツ」ではある。ここでループにまだ慣れない人のために「変数の箱」の演習をやっておこう。

演習 1 ☆ 階乗の PAD 図とプログラムを見ながら、 n として「5」を入力したときの変数の箱の変化を記入せよ (かっこ内の番号も自分で埋めて行くこと)。



- 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。

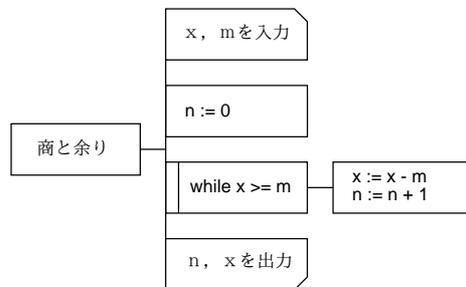


図 3: 商と余り

これは要するに、引ける限り x から m を引き、引けた回数と残った半端を出力すればよいわけである。

```

program sam10a(input, output);
var x, m, n: integer;
begin
  write('x = '); readln(x);
  write('m = '); readln(m);
  n := 0;
  while x >= m do begin x := x - m; n := n + 1 end;
  writeln('shou = ', n:1, ' amari = ', x:1)
end.
  
```

- 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。

この手の問題をプログラムにするには問題文の条件が満たされない間ループする、という形になることに注意。

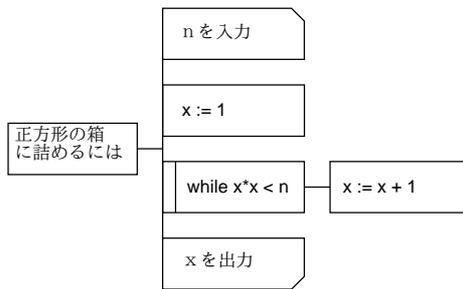


図 4: 箱詰め?問題

```

program sam10c(input, output);
var n, x: integer;
begin
  write('n = '); readln(n);
  x := 1;
  while x*x < n do x := x + 1;
  writeln('x = ', x:1)
end.
  
```

最後に最大公約数だが、せっかく面白い問題だから、PAD だけ示しておくので、できなかった人は自主的に再挑戦してみて欲しい。

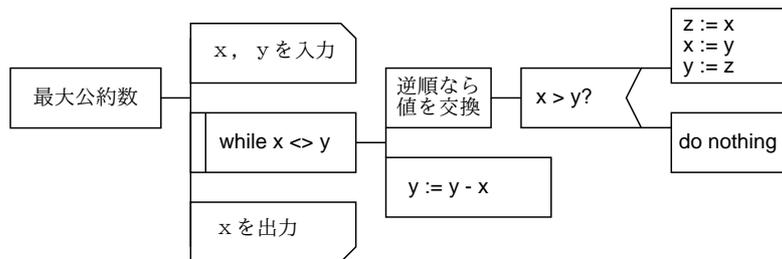


図 5: 最大公約数

2 2分探索法/☆

さて、前回の練習問題では値が整数ばかりだったが、実数を駆使した面白いテーマがあるのでこれを取り上げよう。そのテーマとは、一般に関数 f について「 $f(x) = 0$ 」を満たす x (つまり根) を求めることである。たとえば、 $f(x) = x^2 - 2$ だとすれば、その根は $\sqrt{2}$ ですね? ここで述べる方

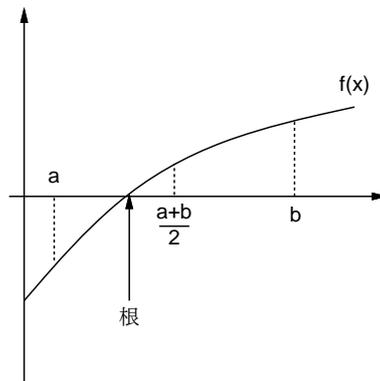


図 6: 根の 2 分探索の原理

法では、関数が (根を含む知られた区間 $[a, b]$ で) 単調増加かつ連続であれば、どんな関数であろうとその根を求めることができる。すごいでしょう? ただし、代数的に ($\sqrt{2}$ とか) 求まるのではなく、数値的に (1.41421356 のように) だが。精度は (計算機の演算命令の限界内で) 自由に指定できる。

その考え方は次の通り。まず、根を含む区間 $[a, b]$ の情報を考える。 $f(x)$ は単調増大なのだから、 $f(a) < 0$ かつ $f(b) > 0$ のはずである。さて、許される誤差を e とすると、 $b - a \leq e$ ならもう答えは求まっていることになる (でしょう?)。もしそうでなければ、区間の中点 $\frac{a+b}{2}$ を考え、ここでの $f(x)$ を計算してみる。それがもし正であれば、新しい区間として $[a, \frac{a+b}{2}]$ を考え (つまり b を $\frac{a+b}{2}$ で置き換える)、そうでなければ、新しい区間として $[\frac{a+b}{2}, b]$ を考える (つまり a を $\frac{a+b}{2}$ で置き換える)。どちらにせよ、新しい区間に根が含まれていることは図 6 を見れば分かると思う。これを 1 回繰り返すごとに区間の幅は半分ずつになっていくから、いつかは幅が e よりも小さくなる。つまり、指定した誤差で解が求まったことになりますね? このアルゴリズムを PAD で記したものを図 7 に示す。

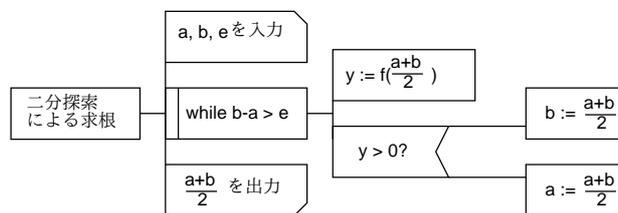


図 7: 根の 2 分探索の PAD

演習 2 ☆ 上の PAD を Pascal に直し、2 の平方根や 3 の立方根を求めて

みよ。(そのためには $f(x)$ を適当な計算式で置き換えて動かす)。なお、変数はすべて実数でないと意味がないことに注意。

3 ループの中での入力と出力/*

これまでのところ、ループの本体には read も write も含まれていなかった。しかし、ループの本体に入出力があると次のようなことができる。

- ループ本体に入力があると、いくつものデータを次々に読み込んで処理することができる。
- ループ本体に出力があると、いくつものデータを次々に出力して、表のようなものを作り出すことができる。

実際見てみよう。まず次の問題を考えてみる。

- a. 実数を次々に入力して、その合計を求めよ。ただし、0 が来たらそこでデータは終わりだとする。

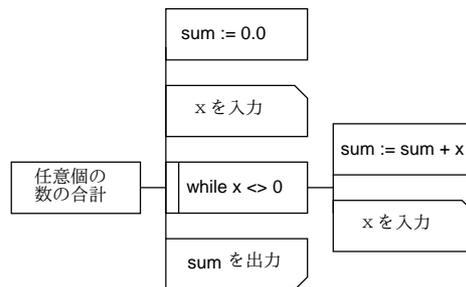


図 8: ループ中に入力のある PAD 図

これを PAD にしたものを図 8 に示す。入力が 2 箇所にあるのがちょっと変わっていますね? これは、while ループの条件でデータが 0 かどうか調べるには、ループの前でまず 1 回データを読み込んでおかなければうまく行かないからである。Pascal にしたものは次の通り。

```
program sam9a(input, output);
var x, sum: real;
begin
  sum := 0.0;
  write('x = '); readln(x);
  while x <> 0.0 do begin
```

```

    sum := sum + x;
    write('x = '); readln(x)
end;
writeln('sum = ', sum:8:3)
end.

```

実行例は次の通り。

```

% pc sam9a.p
% a.out
x = 10.5
x = 2.3
x = 8.9
x = 0
sum = 21.700
%

```

では次の問題も考えてみよう。

- b. 貯金の元金 m 、利率 $r(\%)$ 、目標金額 g を入力し、1 年後つまり 1996 年から目標金額に達するまでの年と、現在高を出力せよ。

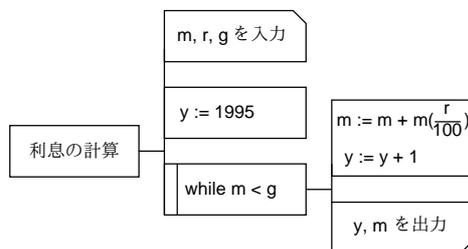


図 9: ループ中に出力のある PAD 図

こんどは入力は 1 回だが、出力は現在高が目標に達するまで繰り返し行うことになる。その PAD を図 9 に示す。Pascal では次の通り。年は「1 つずつ数える」から整数、他は実数の変数になることに注意。

```

program sam9b(input, output);
var y: integer;
    m, r, g: real;
begin
    write('money amount = '); readln(m);
    write('interest rate = '); readln(r);
    write('goal amount = '); readln(g);
    y := 1995;

```

```

while m < g do begin
    m := m + (m*r/100.0); y := y + 1;
    writeln('year ', y:4, ' amount = ', m:8:4)
end
end.

% pc sam9b.p
% a.out
money amount = 100.0
interest rate = 5
goal amount = 150.0
year 1996 amount = 105.0000
year 1997 amount = 110.2500
year 1998 amount = 115.7625
year 1999 amount = 121.5506
year 2000 amount = 127.6282
year 2001 amount = 134.0096
year 2002 amount = 140.7100
year 2003 amount = 147.7455
year 2004 amount = 155.1328
%
```

このように、ループの中に入出力がある場合には、「ある解が求まるまで繰り返す」というのとはだいぶ感じが違って、データ 1 個ごとに繰り返しながら入力や表示を行う、という感覚になる。慣れて頂きたい。

演習 3 Δ a を改良して、でデータの合計だけでなく、平均も求めるようにせよ。

演習 4 Δ b を修正して、目標金額までではなく、2010 年までの各年の金額を打つようにせよ。

4 for ループ—固定回数の繰り返し/*

さて、ごく非常に簡単な問題として、「整数 N を入力し、1 から N まで順に出力せよ」というのを考える。この PAD を図 10 に示す。これはこれでいいのだが、

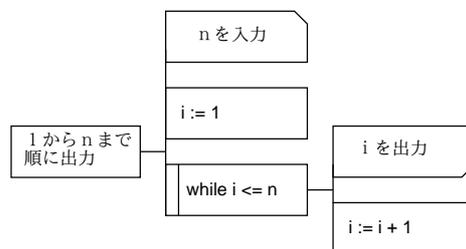
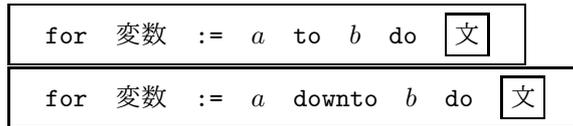


図 10: 固定回数のループを while で

実は計算機のアルゴリズムでは「整数変数を、まずある決まった値 a にし、それから別の値 b に到達するまで1個ずつ増やしながら(または減らしながら)何かをする」というパターンは非常に多く現われる。そのため、このようなパターンの繰り返しには「for ループ」ないし「do ループ」という決まった名前がついていて、さらに多くのプログラム言語はそのための文が専用に用意されている。Pascal でも



という形をした「for 文」が用意されている。もちろん、下の方が「減らしながら」に対応する。(なお、「変数」は整数型でなければならない。また a や b の所には整数型の任意の「式」が書ける。)そして PAD でもこれに対応した書き方をする。それを利用すれば、上の問題の PAD は図 11 のようになる。ひどく簡単でしょう?



図 11: 固定回数のループを for で

演習 5 Δ 上の 2 つの PAD をともに Pascal にして動かせ。

演習 6 Δ 次の問題を PAD で書き、Pascal にして動かせ (for と while のどちらが適しているか考えよう)。

- a. 0 が来るまで次々にデータを読み、最大と最小の値を打ち出す。
- b. 整数 N を読み込み、 $x = 1 \dots N$ について、 x 、 x^2 、および $\frac{1}{x}$ の表を打ち出す。
- c. 整数 N を読み込み、値が N 未満のフィボナッチ数列を打ち出す。¹

5 WWW ページの作り方

5.1 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? 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。

¹フィボナッチ数列とは、1、1、2、3、5、8、13、…のように 3 番目以後の値がその前 2 個の和に等しいような数列。

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 Mosaic を使って WWW を眺めている間に、必要に応じて世界中の WWW サーバに対しインターネット (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。インターネットでは、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.ecc.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、教育計算センターの、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、前にやったように皆様のメールアドレスは

```
ユーザ名@komaba.ecc.u-tokyo.ac.jp
```

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

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

というホストが扱っている。

5.2 WWW サーバと URL/*

さて、WWW の話に戻って、情報棟の WWW サーバ計算機は

```
www.komaba.ecc.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」が入っているのので、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.ecc.u-tokyo.ac.jp/index.html
http://www.komaba.ecc.u-tokyo.ac.jp/~kuno/jouhou95.html
http://www.komaba.ecc.u-tokyo.ac.jp/~kuno/qa/index.html
http://www.komaba.ecc.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.ecc.u-tokyo.ac.jp/
http://www.komaba.ecc.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:」などいくつか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。

5.3 個人のページのありか/☆

いよいよ自分の情報ページを作っていたらごう。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。ページはいくつでも持てるが、それはすべて「WWW/」の下に置く必要がある。ホームページ以外のページは、ファイル名は自由につけていいが、最後は「.html」で終らなければいけない。また、ページの中には GIF 形式の絵が入れられるが、その GIF ファイルも「WWW/」の下に置く必要がある。

演習☆ 「mkdir WWW」を実行して WWW ディレクトリを作成せよ。また、「cp なんか.gif WWW/mark.gif」により、前に作った自分のトレードマークの GIF ファイルを mark.gif という名前で WWW ディレクトリにコピーしておくこと。

5.4 HTML によるページ記述/*

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。最初に HTML の概要を説明し、続いて実際に作って頂こう。

5.4.1 全体的な形

まず、HTML ファイルの全体は次の形をしていないといけない。

```
<HTML>
<HEAD>
<TITLE>タイトル</TITLE>
</HEAD>
<BODY>
  本体
</BODY>
</HTML>
```

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(こういうの「タグ」と呼ぶ)。ファイル全体は<HTML>... </HTML>で囲まないとけない。その中には<HEAD>...</HEAD>という部分と<BODY>...</BODY>という部分があり、<HEAD>...</HEAD>の中には通常<TITLE>○○○</TITLE>というのだけを入れる。ここは、ページ全体のタイトル (Mosaic で見ると Document Title:という部分に表示される) を書く。そして、ページの中身はすべて<BODY>...</BODY>のところに書く。

5.4.2 見出しと絵

ページの中身はだいたい、見出しで始まる。見出しは

```
<H1>○○○○○</H1> --- いちばん大きい見出し
<H2>○○○○○</H2> --- 2レベル目の見出し
<H3>○○○○○</H3> --- 3レベル目の見出し
```

のように、H6 まである。見出しは大きい字で目立つように表示される。見出しの中でもそれ以外の場所でも、

```
<IMG SRC="GIF ファイル名">
```

というのを書いておけば、指定した GIF ファイルの絵をちょうどその場所に埋め込むことができる。にはというタグはない。

5.4.3 パラグラフと箇条書き

本文(地の文)は<P>...</P>で囲んで書く。これをパラグラフとよぶ。パラグラフは何行に渡ってもよく、適当に画面幅で詰め合わせて表示される。

```
<P>わたしは…ほにゃほにゃ…
へげへげ…です。</P>
```

なお、詰合せされては困る場合には<PRE>...</PRE>で囲むように。

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

あと、箇条書きもよく使う。

```
<UL>
<LI> 最初の項目...
<LI> 2番目の項目...
<LI> 3番目の項目...
...
</UL>
```

箇条書の各項目は複数行に渡ってよく、適宜詰め合わされる。

5.4.4 リンク

そして、WWWの最大の特徴である、「クリックするとよそへ飛ぶ」というのは「リンク」と呼ばれるタグで指定する。

```
<A HREF="selfintro.html">自己紹介はここです</A>
<A HREF="/~g400051/index.html">××さんのページはこちら</A>
<A HREF="http://www.komaba.ecc/u-tokyo.ac.jp/">駒場のページ</A>
```

つまり、<A>...で囲む部分が「下線つきでクリック可能なテキスト」になり、飛び先は<A>タグのHREF="..."のところ指定する。同じディレクトリにある(つまり自分の)ページは単にファイルだけ指定すればよい。他人のページは「/ユーザ名/」のところから指定すればよい。それ以外の場合にはさつき説明したURLをすべて指定する。

5.4.5 横線

おまけですが、ページの内容の区切りに横線を引きたい場合には<HR>というタグでできます。

5.5 自分のページを作ろう!/☆

演習☆ ではついに、自分のページを作ることしましょう! Mule を動かし、`~X^F` でファイル名として「`WWW/index.html`」を指定してから、以下の内容をしてきとーにアレンジして打ち込むこと (友人は2~3人適当に選ぶ)。

```
<HTML>
<HEAD>
<TITLE>〇〇's Home Page</TITLE>
</HEAD>
<BODY>
<H1><IMG SRC="mark.gif">〇〇のホームページです</H1>

<P>こんにちは。〇〇です。(その他適当に
ご挨拶を書く) </P>

<H2>私が作っているページ</H2>

<UL>
<LI><A HREF="selfintro.html">自己紹介のページ</A>
<LI><A HREF="aboutcomputer.html">計算機についてどう思うか</A>
</UL>

<H2>私のお友達のページ</H2>

<UL>
<LI><A HREF="/~gxxxxxx/index.html">~さん</A>
<LI><A HREF="/~gyyyyyy/index.html">~さん</A>
</UL>

<HR>
<P>よろしく願います。</P>
</BODY>
</HTML>
```

これを Mosaic で表示すると図 12 のようになるはず。

演習☆ 作り終わったらファイルに保存してから Mosaic で「久野クラス受講生ホームページ一覧」へ行き、自分の名前のあるところをつついてページが表示されることを確認する。さらにそこから他人のページをつついて出してみる (その人が自分のページを完成させた後でないとうまく行きません。当たり前)。

演習☆ ホームページを参考にして、「自己紹介のページ」(ファイル名は `selfintro.html`) と「計算機について思うことのページ」(ファイル名は `aboutcomputer.html`)



図 12: サンプルホームページ

を作れ。自己紹介のページには生まれ、趣味、好きな食べ物は最低含めること。計算機についてのページには好きな内容を書いてよいが、最低 400 字くらいは書いて欲しい。

A 本日の課題 **8A**

本日の課題は WWW のページを作成することです。自己紹介と計算機についてのページまで必ず作る。すでに色々作っている人も、上記 2 つのページは(まだなければ)作ってください。

なお、今後課題提出としてページを作った場合には、久野クラスのニュースグループに「どこにどんなページを作った」かを報告する記事を出してください。私に毎日全員のページを見回りしろというのはご勘弁を。

B 次回までの宿題 **8B**

次回までの課題は「演習 3 か演習 4 の好きな方」および「演習 6 のどれか 1 つ以上」とします。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 **8B** 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「ポイント」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。