

Risa/Asir ドリル, 2010

高山信毅, 野呂正行

2010年(平成22年), 11月6日 版: コメントは
takayama@math.kobe-u.ac.jp または noro@math.kobe-u.ac.jp まで

はじめに

Risa/Asir ドリルは、著者 (T) が徳島大学総合科学部および神戸大学理学部で数学系の学生におこなってきた計算機プログラミングの入門講義および演習をたたきだいにして書いた本である。この講義ではさまざまな言語 C, Pascal, 8086 の機械語, Ubasic, sm1, Mathematica 等を用いて実習してきたが、利用するプログラム言語は異なっても基本的内容は同じであった。ちなみに本書の原稿のおおもとは、1991 年に著者 (T) が学生に配布した、Mathematica の入門テキストである。

2000 年の秋に著者 (N) が富士通研究所より神戸大学へ転職してきたのを機会に、著者 (T) はこの年の計算機プログラミングの入門講義および演習を富士通研究所で著者 (N) が開発にたづさわってきた数式処理システム Risa/Asir を用いておこなうことにした。Risa/Asir は研究用システムとしてすぐれた点を多くもつシステムであったが、教育用途に利用するにはいろいろと不満な点もあった。著者 (N) は、著者 (T) およびいろいろと珍奇なことをやってくれる学生にのせられて Risa/Asir を教育用途にも使えるよういろいろと改造した。実習室用に CDROM を入れるだけで Windows 2000 で起動できる Asir、初心者にはやさしい入力エラーの取扱い、ファイル IO、2 次元簡易グラフィックス、さらには教育用途のために、メモリを直接読み書きする peek, poke まで付けた (ちなみにこれはセキュリティホールになるので普通は利用できない)。この Risa/Asir をつけた実習は著者 (T) がおこなった他のいろいろな言語による実習のなかでも成功の部類にはいるものであり、本として出版してみようという気になったのである。

さて前にもいったようにこの講義および演習は数学科の学生向けであった。講義の目的は以下のとおりである (目標はそもそもなかなか達成できないものであるが...).

1. 高校数学 A, B, C の計算機に関する部分を教えられるような最低限の知識と技術を身につける。
2. 数学科の学生は卒業後、計算機ソフトウェア関連の職業につくことが多いが、その基礎となるような計算機科学の全体的な基礎知識を得る。
3. 計算数学が現代の科学技術のなかでどのように利用されているかおぼろげに理解してもらう。また計算数学が数学のなかの一つの研究分野であることを理解してもらい、とくに計算代数への入門を目指す。
4. 数学を活用する仕事 (含む数学者) についての場合に、数式処理システム等を自由に使えるようにする。

この本では講義の内容に加えてさらに数学的アルゴリズムに関する特論的な内容や、Asir のライブラリプログラムを書くための方法、Asir に C のプログラムを組み込むための方法、計算数学のシステムをたがいに接続する実験プロジェクトである、OpenXM を利用した分散計算法など、オープンソースソフトとしてリスタートした、Risa/Asir のための情報も加筆した。この本が Risa/Asir の利用者、開発参加者にも役立つことを願っている。

なお、この本は数学科の学生向けの講義をもとに書かれたが、一部分を除き大学理系の微分積分学、線形代数学程度の知識があれば十分理解可能である。実際本書は工学系の学生や、高校生向けに利用したこともある。この本がさまざまな人にとり有益であることを願っている。

2002 年 (平成 14 年)10 月、著者

追記: このテキストによる講義例として下記のビデオを公開している。

<http://fe.math.kobe-u.ac.jp/Movies/cm/2006-keisan-1-nt.html>

目次

第 1 章	Asir の操作	9
1.1	login, logout, キー操作 (unix 編)	9
1.2	logon, logoff, キー操作 (Windows 編)	10
1.3	Risa/Asir の起動法と電卓的な使い方	10
1.4	Asirgui (Windows 版)	17
1.5	エディタを利用してプログラムを書く (unix 編)	18
1.6	エディタを利用してプログラムを書く (Windows 編)	21
1.7	参考: Asir について知っておくと便利なこと	21
1.7.1	起動時のファイル自動読み込み	21
1.7.2	デバッガ	22
1.7.3	ファイル読み込みディレクトリの登録 (unix 編)	22
1.8	cfep/asir (MacOS X 用) を コマンドラインで使う	23
1.8.1	GUI とコマンドラインインタフェース	23
1.8.2	準備	24
1.8.3	asir コマンド	25
1.9	章末の問題	26
第 2 章	unix のシェルとエディター emacs	27
2.1	unix のシェルコマンド	27
2.2	Emacs	30
第 3 章	計算機の仕組み	33
3.1	CPU, RAM, DISK	33
3.2	計算機の歴史	35
3.3	2 進数と 16 進数	35
3.4	章末の問題	37
第 4 章	Risa/Asir 入門	41
4.1	Risa/Asir で書く短いプログラム	41
4.2	デバッガ	44
4.3	関数の定義	45
4.4	章末の問題	46
第 5 章	制御構造	49
5.1	条件判断と繰り返し	49
5.2	プログラム例	49
5.3	glib について	58

第 6 章	制御構造とやさしいアルゴリズム	61
6.1	2 分法とニュートン法	61
6.2	最大値と配列	64
6.3	効率的なプログラムを書くには?	67
6.4	章末の問題	68
第 7 章	ユークリッドの互除法とその計算量	71
7.1	素因数分解	71
7.2	計算量	71
7.3	互除法	72
7.4	参考: 領域計算量と時間計算量	74
7.5	章末の問題	76
7.6	章末付録: パーソナルコンピュータの歴史 — CP/M80	76
第 8 章	関数	81
8.1	リストとベクトル (配列)	81
8.2	関数と局所変数	82
8.3	プログラム例	85
8.4	デバッガ (より進んだ使い方)	90
8.4.1	ブレークポイント, トレースの使用	90
8.4.2	実行中断	92
8.5	章末の問題	94
第 9 章	常微分方程式の数値解	97
9.1	差分法	97
9.2	不安定現象	99
9.3	解の収束定理	99
9.4	例	102
第 10 章	入出力と文字列の処理, 文字コード	105
10.1	文字コード	105
10.1.1	アスキーコード	105
10.1.2	漢字コードと ISO2022	106
10.1.3	全角文字と ¥ 記号	109
10.2	入出力関数	109
10.3	文字列の処理をする関数	110
10.4	ファイルのダンプ	110
10.5	章末の問題	112
第 11 章	数の内部表現	117
11.1	peek と poke	117
11.2	32 bit 整数の内部表現	120
11.3	整数の内部表現	121
11.4	浮動小数点数の内部表現	121

第 12 章 再帰呼び出しとスタック	125
12.1 再帰呼び出し	125
12.2 スタック	127
第 13 章 リストの処理	133
13.1 リストに対する基本計算	137
13.2 リストと再帰呼び出し	139
第 14 章 整列：ソート	141
14.1 バブルソートとクイックソート	141
14.2 計算量の解析	142
14.3 プログラムリスト	142
14.4 ヒープソート	144
14.4.1 ヒープ	145
14.4.2 ヒープの配列による表現	146
14.4.3 downheap()	146
14.4.4 ヒープソート	148
14.5 章末の問題	149
第 15 章 1 変数多項式の GCD とその応用	153
15.1 ユークリッドのアルゴリズム	153
15.2 単項イデアルと 1 変数連立代数方程式系の解法	153
15.3 計算効率	156
第 16 章 割算アルゴリズムとグレブナ基底	161
16.1 Initial term の取り出し	161
16.2 多項式の内部表現と initial term の取り出しの計算効率	162
16.3 割算アルゴリズム	165
16.4 グレブナ基底	167
16.5 グレブナ基底と多変数連立代数方程式系の解法	172
第 17 章 RSA 暗号系	177
17.1 数学からの準備	177
17.2 RSA 暗号系の原理	178
17.3 プログラム	179
第 18 章 構文解析	185
18.1 再帰降下法	185
18.2 プログラム minicomp.rr	186
18.3 LR パーサ	190
第 19 章 OpenXM と分散計算	197
19.1 OpenXM Asir server	197
19.2 Quick Sort	198
19.3 Cantor-Zassenhaus アルゴリズム	200

第 20 章 行列	203
20.1 行列	203
20.1.1 行列の生成, 要素への代入, スカラー倍	203
20.1.2 行列, ベクトル演算	204
20.2 ガウス消去と連立代数方程式の求解	204
20.2.1 行簡約形と階数	204
20.2.2 行簡約と連立線形方程式の求解	205
20.2.3 ガウス消去アルゴリズム	206
20.3 LU 分解	208
第 21 章 Asir 用のライブラリの書き方	215
21.1 ライブラリの書き方	215
21.1.1 関数名の衝突の回避	215
21.1.2 変数名の衝突	216
21.2 モジュール機能	216
21.3 マニュアルの書き方 その 1: texinfo で直接書く	218
21.3.1 構成	224
21.3.2 書体指定	224
21.3.3 箇条書き	225
21.3.4 例の表示	225
21.3.5 GNU info, HTML に関する指定	225
21.3.6 T _E X による整形	225
21.3.7 GNU info ファイルの生成	226
21.3.8 HTML ファイルの生成	226
21.4 マニュアルの書き方 その 2: コメントから自動生成する	226
第 22 章 Risa/Asir に C で書いたプログラムを加えるには?	227
22.1 asir2000 の source tree	227
22.2 builtin ディレクトリ	228
22.3 例 — 一バイト書き出し関数の追加	228
22.3.1 ファイルの選択	228
22.3.2 関数名, 仕様を決める	228
22.3.3 open_file の改造	228
22.3.4 put_byte の追加	230
22.4 error 処理	232
22.5 プログラムの解読	232
第 23 章 Mathematica と Asir	233
第 24 章 常微分方程式の数値解 (C 言語版)	235
24.1 Risa/Asir プログラムのための C 言語入門	235
24.2 Risa/Asir プログラムのための X Window プログラム入門	237
24.3 差分法	237
24.4 不安定現象	240
24.5 解の収束定理	242

24.6 例	244
第 25 章 入出力と文字列の処理, 文字コード (C 言語版)	251
25.1 文字コード	251
25.1.1 アスキーコード	251
25.1.2 漢字コードと ISO2022	252
25.1.3 全角文字と ¥ 記号	255
25.2 ファイルのダンプ	255
25.3 入出力関数	257
25.4 文字列の処理をする関数	258
25.5 章末の問題	258
第 26 章 付録: エラー	263
26.1 Windows のエラー	263
26.2 Asir のエラー と Q & A	263
第 27 章 付録: パス名	265
27.1 ファイル名の付け方	265
27.2 ディスクのなかにどんなファイルがあるのか調べたい	265
27.3 階層ディレクトリ	265
27.4 ドライブ名	267
27.5 自分はどこにファイルをセーブしたの?	268
27.6 Q and A	268
第 28 章 付録: 実習用 CD の利用法	271
28.1 CD の構成	271
28.1.1 asirgui の起動	271
28.1.2 Meadow のインストール (Windows)	271
28.1.3 その他のファイル	271
28.2 Q and A	272
28.2.1 ハードディスクにコピーして利用したい	272
28.2.2 ... プログラムの製造元に連絡して下さい. ENGINE のページ違反です. モジュール ENGINE.EXE アドレス 0137:2004037a9	272
第 29 章 付録: ソースコードの入手	275
29.1 著作権情報	275
29.2 ソースの所在地	276
29.3 マニュアルの所在地	277
索引	281

第26章 付録: エラー

26.1 Windows のエラー

疑問 26.1 Windows の反応がとってもおそくなったり、グラフをかいていたら突然止まったり、メモリーがないから書けないよ、とってきたりする。どうしてか？

答え 26.1 RAM (メモリ) の不足でこのようなことがおこる。とりあえず、止められるなら計算を中断して、マシンを再起動して、再計算してみる。それでも同様な状態になるなら、実際にメモリが不足しているということなので、(お金に余裕のある人は) RAM を買い足せばよい。

ただし、計算を数時間続けてメモリが足りない事態に至るときは、アルゴリズムの本質的改良がないと解けない場合がおおい。計算量の議論を思いだそう。

26.2 Asir のエラー と Q & A

疑問 26.2 ; `RETURN` を入力しても Asir のプロンプトが戻って来ない。返事がない。なぜか？

答え 26.2 " 記号や ' 記号は組にして使用する。たとえば `print("Hello");` としていて、`Hello` のあとの " を忘れていた可能性がある。そのときは " ; `RETURN` でプロンプトが戻るはずである。

{ と } も組にして使用する。これが閉じていないとやはりプロンプトがもどってこないことがある。そのときは } `RETURN` でプロンプトが戻るはずである。または、`CTRL+C` を入力して、プロンプトにもどしてもよい。

疑問 26.3 次のプログラムを実行したらいつまでも終わりません。 ; `RETURN` を入力してもプロンプトがでません。

```
S=0$
for (I=0; I<100; I+1) {
  S = S+I;
}
print(S)$
```

答え 26.3 I の値が更新されない無限ループにおちいってプログラムの実行中です。 `CTRL+C` でプログラムを中断して下さい。

`for (I=0; I<100; I+1)` は誤りで、`for (I=0; I<100; I=I+1)` か `for (I=0; I<100; I++)` に訂正して下さい。

さて、Asir の言語処理系からのエラーメッセージをみよう。

疑問 26.4

```
[346] fctr(x^2-@i*y^2);
fctrp : invalid argument
return to toplevel
```

答え 26.4 取り扱えない引数があたえられている (invalid argument). 有理数を係数とする多項式しか因数分解できない. $x^2 - iy^2$ は複素数を係数としている.

疑問 26.5 次のプログラムの動作が変です.

```
def foo(M) {
  G1 = x+y;
  G2 = x-y;
  G = GM;
  return(G^3);
}
```

答え 26.5 このプログラムはもしかして,

```
def foo(M) {
  if (M == 1) {G = x+y;}
  if (M == 2) {G = x-y;}
  return(G^3);
}
```

というつもりで書いたのではないのでしょうか? GM は引数 M の値に応じて, G1, G2 と解釈されるわけではなくて, GM という新しい局所変数と解釈され, 初期値は 0 となります.

疑問 26.6 asirgui の画面で, マウスを click してそこにある入力を修正しようとしてもできません.

答え 26.6 asirgui は Windows 標準のユーザインタフェースにしがっていない. むしろ, unix のターミナルに似せてつくってある. たとえば, Windows 版 asirgui は, 通常の Windows における慣習とは異なる形のコピーペースト機能を提供している. Window 上に表示されている文字列に対しマウス左ボタンを押しながらドラッグすると文字列が選択される. ボタンを離すと反転表示が元に戻るが, その文字列はコピーバッファに取り込まれている. マウス右ボタンを押すと, コピーバッファ内の文字列が現在のカーソル位置に挿入される. 既に表示された部分は readonly であり, その部分を改変できないことに注意して欲しい.

疑問 26.7 $2\sin 2\pi$ の近似値を計算しようとして次のように書いたらエラーがでました.

```
deval(2sin 2@pi);
```

答え 26.7 かつこおよび掛け算記号 * が必要です.

```
deval(2*sin(2*@pi));
```

疑問 26.8 2 次関数の軸と頂点を覚えて沢山表示したい. どうすればいいのか?

答え 26.8 次のような関数を作り引数の値を変化させればよい.

```
def drawquad(A,B) {
  x^2+A*x+B のグラフを描く.
}
```

疑問 26.9 `X = deval(sin(3 .14));` エラーがでます. どうしてですか?

答え 26.9 `3` と `.14` の間に空白があります. 空白は文と文の間, 変数と演算子の間, など自由に入れることができますが, ひとまとまりの数を空白では区切れません.

疑問 26.10 フィードバック用紙に次のように書いたら書き方が不適切といわれました. どう書いたらいいですか?

1. 記号の意味がよくわからない.
2. みなれない記号があつてむつきしそうな感じがした.
3. プログラミングの仕方がよくわからない.
4. 何がエラーがよくわからない.

答え 26.10

1. “記号の意味がよくわからない.” では説明のやりようがないので, たとえば “記号 `2*pi @`” の意味がよくわからない具体的に書く.
2. “みなれない記号があつてむつきしそうな感じがした” では漠然とした感想でこれ以上議論が進まない. たとえば “記号 `++`, `<=` がむつきしそうに見える” などと具体的に書く.
3. “プログラミングの仕方がよくわからない” ではやはり説明のやりようがないのでたとえば “p.99 のプログラムがよくわからない. 特に 5 行目.” とやはり具体的に書く.
4. “何がエラーがよくわからない.” も具体的にエラーの前後の行を書き, エラーメッセージもそのまま写しておく.

疑問 26.11 `cfep/asir` のグラフィック画面にあるバーは何ですか?

答え 26.11 3 次元グラフィックスを表示した時に, 視点の位置を変えるためのものです. `cfep/asir` のサンプルプログラムのなかの `ray.rr` を実行してみてください.

疑問 26.12 `glib3.rr` って何ですか?

答え 26.12 `glib_line` などの関数を定義しています. この関数の中ではより原始的な `graphic` 命令を用いて線を描画しています.

疑問 26.13 括弧のう忘れをよくします.

答え 26.13 `cfep/asir` では, 括弧ボタンをおせば, 括弧の対応付けを調べてくれます. 押すたびに括弧の範囲が色を変えて表示されます. `emacs` では括弧の対応付けは入力時に調べてくれます. 右括弧を入力すると対応する左括弧にカーソルが一瞬飛びます.

疑問 26.14 文字列ってなんですか?

答え 26.14 asir が扱う代表的データには 数字, 文字列, リストがあります. 変数にはこれらの何をいれてもいいです. 文字列は " で囲ったものです.

例:

```
X="good";
Y=" morning";
Z=" evening";
print(X+Y);
print(X+Z);
```

文字列の足し算は文字をくっつける操作です. 引き算やかけ算は定義されていません.

疑問 26.15 X11 って何ですか?

答え 26.15 X は 1984 年、マサチューセッツ工科大学で開発され、現在のバージョンである X11 は 1987 年 9 月に登場した. これは GUI 構築のための基本ソフトウェアを提供するものであり、多くの unix 系のシステムで利用されている. MacOS にはオプションとして標準装備されている.

疑問 26.16

```
load("~/abc.rr");
```

と入力してもホームの下の abc.rr を読み込めません.

答え 26.16 現在の asir ではシェルでホームディレクトリを意味する ~ は使えません (講義での説明ミス). 次のように入力するとホームの下の abc.rr を読みこめます.

```
load(getenv("HOME")+"/abc.rr");
```

疑問 26.17 次のプログラムは cfep/asir で実行できますが、コマンドライン版の load コマンドで実行しようとすると nsum undefined とエラーになります.

```
nsum(10);
def nsum(N) {
  S = 0;
  for (I=1; I<=N; I++) {S = S+I;}
  return(S);
}
```

答え 26.17 nsum を呼び出す前に def で nsum の定義をして下さい. 上の例では nsum(10); を最後の行に移動. cfep/asir の時は例外的にあとで定義しても動作します. (if (1) { } としているため.)

疑問 26.18 Windows でファイルをうまく load できません.

答え 26.18 load コマンドの仕組みは Windows では少々複雑です. ファイル, 読み込みのメニューを利用して下さい.

疑問 26.19 テキストエディタって何ですか?

答え 26.19 Windows だとメモ帳 (notepad), Sakura editor, 秀丸, Meadow (emacs) 等がテキストエディタです. Mac だとテキストエディットをフォーマット, 標準テキストにする, で使った場合, carbon emacs 等がテキストエディタです. 詳しくは文字コードの章を見て下さい.

疑問 26.20 コマンドラインの asir を使おうとしてますが, 次のようなエラーがでます. 何が間違ってますか?

```
bash-2.05b$ load("./Desktop/abc.rr");
bash: syntax error near unexpected token './Desktop/abc.rr'
bash-2.05b$ 1+2;
bash: 1+2: command not found
```

答え 26.20 これは unix シェルに asir のコマンドを入力しているからです (asir 語のわからない人に asir 語を話しかけている状態). コマンドラインインタフェースでは自分は今どのソフトにコマンドを入力しているかを常に把握している必要があります. 同じ窓に入力していても, 応答しているソフトウェアは変わっています. 端末ウィンドウは起動時には unix シェルがまず動作します (unix シェルはシェル語しか理解しません). たとえば ls (ファイル一覧の表示) はシェルのコマンドです. これをコマンドラインの asir に入力すると

```
[1365] 1+2;
3
[1366] ls ;
ls
[1367]
```

みたいになります.

疑問 26.21 load コマンドってなんでしたっけ? load コマンドと import の違いは?

答え 26.21 load も import も指定したファイルからプログラムを読み込みます. 索引から load を検索して下さい. import は一度読み込んだことのあるファイルの再度の読み込みはしませんが, load コマンドは実行毎にファイルを読み込みます. たとえば, システムが提供しているライブラリプログラム (glib3.rr など) の読み込みには import, 自分で書いてまだ修正中のプログラムの読み込みには load を使います.

疑問 26.22 taylor 展開のプログラムで x が大文字でなくて小文字です. なぜですか?

答え 26.22 小文字は asir が多項式の計算をするときに不定元として使われます. このプログラムではまず Taylor 展開を多項式として計算しています.

疑問 26.23 cfep で入力した文字が勝手に大きくなってます.

答え 26.23 これでも害はないですが, フォントメニューで字を小さくできます.

疑問 26.24

```
T1=time(); print(T1[0]-T0[0]);
```

を入れると, array or list expected (リストでも行列でもないの...) とエラーがでます.

答え 26.24 実行前の `T0=time()`; を忘れていていると思います. `T0` には `0` がはいたままなので, `T0[0]` がエラーをおこします.

疑問 26.25 `glib_putpixel(X,Y)` ってなんですか?

答え 26.25 座標 (X,Y) に点を打つ命令です. 詳しくは索引でこのコマンドを参照. google で検索する場合は `asir glib_putpixel`.

疑問 26.26 `return` と `print` の違いがわかりません.

答え 26.26 例で説明します.

```
def ddd(N) {
    return(2*N);
}
def ddd2(N) {
    print(2*N);
}
A=ddd(3);
B=ddd2(3);
```

このとき `A` には `6` が代入されますが, `B` には `0` が代入されます. `ddd2` の実行の時に `6` が画面にプリントはされます.

疑問 26.27 次のプログラムを実行するといつまでも終了しません.

```
import("glib3.rr");
glib_open();
glib_window(0,10,0,100);
for (K=0; K<10.0; K = K+0,03) {
    glib_putpixel(K,K^2); glib_flush();
}
```

答え 26.27 `K=K+0,03` の部分が `K=K+0.03` の間違いです. コンマとピリオドの違いに注意. いまの場合, コンマは文の区切りとしてあつかわれているので, `K=K` が実行されているだけで, `K` が増えなくて無限ループになっています. `print(K)` を挿入して確かめると良い.

疑問 26.28 Windows 版で実行ログをとりましたが, それを開くことができません.

答え 26.28 ファイル名に `.txt` の拡張子をつけて下さい.

“Risa/Asir ドリル” はオンライン書籍ですので, 入手法が普通の書籍とは異なります. この本を引用する場合は, 次のようにお願いいたします.

簡単版:

野呂正行, 高山信毅 著: Risa/Asir ドリル, 2010 (入手については
<http://www.math.kobe-u.ac.jp/Asir/index-ja.html> を参照)

詳細版:

野呂正行, 高山信毅 著: Risa/Asir ドリル, 2010
Risa/Asir を用いた (数学) プログラミングの入門書.
<http://www.math.kobe-u.ac.jp/Asir/index-ja.html> より PDF, PS 版を入手可能. 配布, 複製
は自由.

索引

- ;, 13, 51
- ==, 43
- ?:, 168
- \$, 12, 51
- %, 66, 71
- &&, 43
- <=, 14, 43
- \ (バックスラッシュ), 26
- 0 次元イデアル, 172
- 16 進数, 35
- 2 進数, 33, 35
- 2 進展開, 182
- 2 重ループ, 55
- 2 分法, 62
- 32 bit 整数, 120
- 8.3 形式, 35, 265

- 8080, 34, 35, 76
- 8086, 35, 39

- allocatemem, 70, 173
- append, 137
- argument, 82
- array, 81
- asciitostr, 94, 110
- Asir Contrib, 215
- asir_assert, 232
- asirgui, 264, 271
- asirrc ファイル, 21, 199

- basic, 76
- bignum, 119–121
- blacklisted block, 157
- break, 49, 52
- Buchberger アルゴリズム, 161
- Buchberger アルゴリズム, 168
- byte, 33, 35

- Cantor-Zassenhaus アルゴリズム, 200
- car, 137
- cd, 26, 265, 267
- cdr, 137
- chmod, 26
- clone_vector, 139
- close_file, 110
- coef, 155
- cons, 138
- cp, 26
- CP/M80, 35, 76, 265
- cpmemu, 76
- CPU, 33
- cputime, 57
- ctrl, 13, 215
- cvsup, 277
- C 曲線, 126
- C 言語, 227

- debug, 44
- def, 45
- define 文, 51
- deg, 155
- deval, 13, 44, 121
- diff, 68, 95
- dir, 265
- dn, 68
- do-while, 49
- dp_ptod, 165

- emacs, 28
- error, 28
- EUC コード, 106, 252
- eval, 44, 64
- eval_str, 110
- extern, 84, 216

- false, 43

- fclose, 256
- fctr, 12
- fep, 28
- fgetc, 256
- FIFO, 127
- flist, 15, 215
- fopen, 256
- for, 14, 41, 49, 51
- Fourier 展開, 55
- fputc, 256
- FreeBSD, 35

- GCD, 71, 200
- get_addr, 117
- get_byte, 110, 228
- get_line, 109
- gets, 257, 258
- glib_line, 52
- glib_open, 52
- glib_putpixel, 52
- glib_window, 52
- goto-line, 28
- gr, 156, 170

- help, 15
- hex_dump, 117

- iand, 111
- IBM PC コンパチ機, 34, 35
- IBM PC 互換機, 34, 35
- idiv, 68, 71
- IEEE754, 121
- if, 42, 49, 50
- in_>(*f*), 162
- initial term, 161, 162
- interrupt, 12, 93
- inv, 178, 179
- invalid argument, 264
- ishift, 118
- ISO2022, 106, 252

- JIS コード, 106, 252

- Kobe Distribution, 276

- last, 26

- length, 88, 137
- lexicographic order, 161
- load, 14, 18, 21, 29
- load(glib), 21
- Lorentz 方程式, 103, 245
- LR パーサ, 190
- ls, 25, 265

- Macintosh, 266
- maedow, 28
- makeinfo, 219
- member, 139
- mkdir, 26, 266
- mod, 177
- module, 216
- more, 25
- MSDOS, 35, 265
- mule, 28

- newmat, 82
- Newton's method, 61
- Newton の運動方程式, 97, 238
- Newton 法, 61
- newvect, 66, 81, 89
- nkf, 107, 226, 254
- nm, 68
- ntoint32, 120

- O-記法, 71
- open, 110, 256
- open_file, 110, 228
- OpenXM, 197, 277
- OpenXM/Risa/Asir, 17, 215, 271
- OpenXM/Risa/Asir, 17
- OpenXM 版 Risa/Asir, 17, 271
- ox_asir, 197
- ox_launch, 197
- ox_pop_local, 198
- ox_rpc, 197
- ox_select, 198
- ox_shutdown, 197

- pari(sqrt), 37
- passwd, 26
- peek, 117

- plot, 12, 53
- poke, 117
- print, 15, 45
- printexpr, 232
- ps, 26
- ptozp, 157
- p_true_nf, 170
- purge_stdin, 109
- put_byte, 110, 228
- pwd, 26, 267

- quick sort, 198
- quit, 25, 45

- random, 66
- red, 68
- reducible, 166
- reduction アルゴリズム, 161
- return, 51, 82
- Risa/Asir CD, 271
- rm, 26
- rmdir, 267
- RSA 暗号系, 177
- rtostr, 110

- scanf, 257
- setprec, 64
- size, 88
- sqrt, 13
- \sqrt{x} , 62
- sscanf, 258
- ssh, 25
- stack overflow, 70, 173
- strtoascii, 110
- subst, 54
- S 多項式, 168

- Taylor 展開, 54, 63, 97, 238
- texi2html, 219
- texinfo, 218
- true, 43
- type, 82, 139, 165

- unix, 31, 35

- vtol, 82

- w, 25
- while, 49
- Windows, 35

- Z80, 35

- アスキーコード, 105, 179, 251
- アドレス空間, 34
- 位数, 177
- 一次不定方程式, 155
- イデアル, 153
- イデアルメンバシップアルゴリズム, 161
- 因数分解, 12, 200
- インストール, 277
- インデント, 51
- 運動方程式, 97, 238
- エスケープコード, 94
- エスケープシーケンス, 94
- オプション引数, 58
- オリジナル版 Risa/Asir, 17
- 改行コード, 106, 252
- 階層ディレクトリ, 265
- 鍵生成, 183
- 拡張子, 35, 265
- 加速度, 97, 238
- かつ, 43
- カレントディレクトリ, 267
- カレントドライブ, 267
- 漢字コード, 106, 252
- 関数, 45, 81, 125
- 機械語, 34
- 強制振動, 98, 239
- 共通零点, 153
- 局所変数, 82, 88
- 偽, 43
- 逆ポーランド記法, 185
- 行列, 82
- クイックソート, 141
- 空リスト, 137
- 繰り返し, 14, 41, 49
- クリック, 9
- クロック, 74
- グレブナ基底, 156, 161
- 群, 177

- 計算量, 71, 142
- 公開鍵, 179
- 後置記法, 128, 185
- 構文図, 185
- コマンドヒストリ, 28
- コマンドプロンプト, 265
- コメント, 51
- 誤差, 63
- 誤差の爆発現象, 99, 242
- 互除法, 72
- 互除法の効率, 156
- 再帰, 129, 139, 200
- 再帰降下法, 185
- 再帰表現多項式, 165
- 再帰呼び出し, 125
- 最大公約数, 71, 179
- 先いれ, 先だし, 127
- サブルーチン, 81
- 差分化, 98, 239
- 差分スキーム, 98, 239
- 差分法, 97, 237, 238
- 差分法の解の収束定理, 99, 242
- 差分法の不安定現象, 99, 240
- シェル, 25
- シフト, 118
- シフト JIS コード, 106, 252
- 終端記号, 186
- しらみつぶし探索, 52
- 真, 43
- 字下げ, 51
- 辞書式順序, 161
- 自動読み込み, 21
- 条件判断, 49
- 条件分岐, 42
- スコープ, 83
- スタック, 127
- セーブ, 28
- セル, 138
- 全角, 109, 255
- ソースコード, 271, 275
- 素因数分解, 137, 179
- 大域変数, 84, 216
- 達人, 41, 117
- 単位元, 177
- 単項生成, 153
- 単振動, 97, 238
- 代数方程式の解法, 64
- ダブルクリック, 9
- ダンプ, 110, 256
- 中断, 12, 26, 93
- 中置記法, 185
- 著作権, 275
- 通分, 68
- 手続き, 81
- データの型, 82
- ディレクトリ, 26, 265, 266
- デバッグ, 44, 90
- トークンへの分解, 188
- トレース, 90
- ドライブ名, 267
- 2分木, 145
- ニュートン法, 61
- 配列, 64, 81
- 半角, 109, 255
- バイト, 33
- バブルソート, 141
- 番地, 33, 34
- パス, 26, 265
- パス名, 265
- ヒープソート, 144
- 引数, 46, 52, 82
- 非終端記号, 186
- ヒストリ, 28
- 秘密鍵, 179
- 開く, 58, 110, 256
- ファイルエクスプローラ, 265
- ファイル識別子, 110, 256
- ファイルの読み込み, 14, 18
- ファイル名, 35, 265
- フィボナッチ数列, 73
- フォルダ, 265, 266
- 複製, 89
- 富士通研究所, 275
- 浮動小数点数, 13, 121
- 浮動小数点数への変換, 44
- フラクタル, 126
- フルパス名, 267
- ブレークポイント, 90

分散計算, 198
分散表現多項式, 165
文法, 185
プロンプト, 25
平方根, 13
変数, 34
変数名, 13
巾の計算, 182
ベクトル, 64, 81, 88
マシン語, 34
または, 43
無平方多項式, 200
メモリ, 33, 44, 83, 117, 125, 163
メモリが足りない, 263
モジュール機能, 216
戻り値, 82
ユークリッド互除法, 72
ユークリッドの互除法, 153
ライブラリ, 215
リスト, 81, 88, 137, 230
ルートディレクトリ, 267
ロングファイル名, 265
ワード, 121
割算アルゴリズム, 161
割算定理, 153