第1章　その他 (Etcetra)

1.1　ファイルの入出力 (InputOutput)

[[[解説]]]

自作した animation をプレゼン資料に貼り付けたり，測定データなどを読み込んで Maple で手軽に表示，加工したくなります．このとき必要となるファイルとのやりとりを紹介します．

1.1.1　ファイル名の取得

ファイル名の取得は，Java の標準関数を使った Maplet パッケージの GetFile 関数を使う．GetFile 関数を呼びだして開いたファイル選択ウィンドウでファイルを指定するとファイルのパスが file1 に入る．

```
> restart; with(Maplets[Examples]): file1:=GetFile();
```

```
"/Users/bob/MapleTest/data1.txt"
```

Windows では”ẽ”/”に変換する必要がある．日本語のファイル名は文字化けして使えない．

```
> with(StringTools): file2:=SubstituteAll(file1,"\\\\","/");
```

```
"/Users/bob/MapleTest/data1.txt"
```

ファイル名の変更は手でやるか，あるいは Substitute 関数を使う．

```
\begin{MapleInput}
> with(StringTools): file2:=Substitute(file1,"data1","data2");
```

```
"/Users/bob/MapleTest/data2.txt"
```

1.1.2　簡単なデータのやりとり

ファイルとの単純なデータのやりとりは writedata,readdata 関数を使う．例えば，以下のようなデータ (T) を作ったとする．

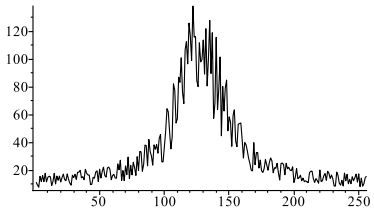
```
> f1:=t->subs({a=10,b=40000,c=380,d=128},a+b/(c+(t-d)^2) ):
> T:=[seq(f1(i)*(0.6+0.8*evalf(rand()/10^12)),i=1..256)]:
```

これをファイル (file1) へ書きだすには，以下のようにする．

```
> writedata(file1,T);
```

読み込んで表示させてみる．readdata の option(=1) は一列のデータを読み込むことを意図している．

```
> T:=readdata(file1,1):
> with(plots): listplot(T);
```



1.1.3　少し高度なデータのやりとり

writeto 関数で出力を外部ファイルへ切り替えることも可能.

```
> interface(quiet=true);
  writeto(file2);
  for i from 1 to 10 do
    s1:=data||i;
    printf("%10.5f %s\n",evalf(f1(i)),s1);
  end do:
  writeto(terminal):
  interface(quiet=false);
```

```
                              false
                               true
```

C 言語の標準的なデータ読み込みに似せた動きもできる．以下は fopen, readline, sscanf, fclose を使ったデータの入力.

```
> fd:=fopen(file2,READ);
  for i from 1 to 2 do
    l1:=readline(fd);
    d:=sscanf(l1,"%f %s");
  end do;
  fclose(fd):
```

```
                               1
                    "  12.42292 data1"
                  [12.42292, "data1"]
                    "  12.46063 data2"
                  [12.46063, "data2"]
```

fd にファイル識別子 (file descripter) を持っていき，readline で 1 行ずつ l1 に読ませる．これを sscanf で format にしたがって d に格納していく．d には自動的に適切な形式で変数を入れてくれる.

```
> d[1]; whattype(d[1]);
```

```
                             12.46063
                              float
```

なお，C 言語と違って，配列の最初を指す index は”1”であることをお忘れなく.

1.1.4 animation の出力

animation などの gif 形式の plot を外部ファイルへ出力して表示させるには，以下の一連のコマンドのようにする．

```
> plotsetup(gif,plotoutput=file2):
> display(tmp,insequence=true);
> plotsetup(default):
```

こいつを quicktime などに食わせれば，Maple 以外のソフトで動画表示が可能となる．3 次元図形の標準規格である vrml も同じようにして作成することが可能です (?vrml; 参照)．

1.1.5 Maple のフィルターとしての利用法

linux 版や Mac 版では文字ベースの maple を使って，filter として高度な作業をさせることが出来ます．スクリプトの中に外部ファイルとの入出力を組み込めば，いままで紹介してきた複雑な動作をブラックボックスの内部処理としてそのまま使えます．

```
[bob@asura0 ~/test]$ cat test.txt
T:=readdata("./data101");
interface(quiet=true);
writeto("./result");
print(T[1]);
writeto(terminal);
interface(quiet=false);
```

とすれば，data101 から読み込んだデータに何らかの処理を施した結果を result に打ち出すことが可能．interface(quiet=true) で余計な出力を抑止しています．これを maple に食わせると

```
[bob@asura0 ~/test]$ /usr/local/maple9.5/bin/maple < test.txt
    |\^/|     Maple 9.5 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2004
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> T:=readdata("./data101");
                            T := [1.23, 2.35]
> interface(quiet=true);
                                 false
                                  true
> quit
bytes used=211000, alloc=262096, time=0.00
```

めでたく出力されているはず．

```
[bob@asura0 ~/test]$ cat result
1.23
```

Mac 版でのパス (path) は下記を参照．

```
bob%  /Library/Frameworks/Maple.framework/Versions/15/bin/maple
    |\^/|     Maple 15 (APPLE UNIVERSAL OSX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2011
 \  MAPLE  /  All rights reserved. Maple is a trademark of
```

```
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
> quit
memory used=1.2MB, alloc=1.4MB, time=0.07
```

1.2 for-loop の基本技 (for-loop2)

[[[解説]]]

1.2.1 ランダムな配列の生成

1 から 100 までの整数 5 個からなる配列の生成.

```
> restart:
  roll:=rand(1..100):
  n:=5:
  A:=[seq(roll(),i=1..n)];
```

$$[93, 45, 96, 6, 98]$$

1.2.2 要素数の取り出し

for-loop で配列を使うときには，配列の大きさ (要素数) が for-loop の終了条件になることが多い．リスト構造では単純に nops とすればよい.

```
> nops(A);
```

$$5$$

1.2.3 すべての要素の表示

配列はおなじ箱が沢山用意されていると考えればよい．配列を for-loop で使うときは，箱を指す数 (示数，index) をいじっているのか，箱の中身 (要素) をいじっているのかを意識すれば，動作を理解しやすい.

```
> for i from 1 to n do
    print(i,A[i]);
  end do;
```

$$1, 93$$
$$2, 45$$
$$3, 96$$
$$4, 6$$
$$5, 98$$

逆順の表示

```
> for i from n by -1 to 1 do
    print(i,A[i]);
  end do;
```

```
                                    5, 98
                                     4, 6
                                    3, 96
                                    2, 45
                                    1, 93
```

逆順の表示 2

```
> for i from 1 to n do
    print(n-i+1,A[n-i+1]);
  end do;
```

```
                                    5, 98
                                     4, 6
                                    3, 96
                                    2, 45
                                    1, 93
```

1.2.4 和

```
> sum1:=0:
  for i from 1 to n do
    sum1:=sum1+A[i];
  end do:
  sum1;
```

```
                                    338
```

課題：積を求めよ.

1.2.5 値の代入

```
> k:=64:
  for i from 1 to n do
    A[i]:=A[i]/k;
  end do:
  A;
```

```
                    [93/64, 45/64, 3/2, 3/32, 49/32]
```

課題：先の和と組み合わせて，全要素の和が 1 になるように規格化せよ.

課題：配列 B へ逆順に代入せよ.

1.2.6　一桁の整数 5 個から 5 桁の整数を作る

まず，一桁の整数でできるランダムな配列を作成する．

```
> roll:=rand(0..9): n:=5: A:=[seq(roll(),i=1..n)];
```

$$A := [3, 5, 4, 0, 7]$$

```
> sum1:=0;
  for i from 1 to n do
    sum1:=sum1*10+A[i];
  end do:
  sum1;
```

$$0$$
$$35407$$

課題：上記と同様にして，10 桁の 2 進数を 10 進数へ変換せよ

1.2.7　255 以下の 10 進数をランダムに生成して，8 桁の 2 進数へ変換せよ．

```
> n:=8: 2^n;
```

$$256$$

```
> roll:=rand(0..255):
  B:=roll();
```

$$161$$

ちょっとカンニング．

```
> convert(B,binary);
```

$$10100001$$

```
> A:=[]:
  for i from 1 to n do
    A:=[irem(B,2),op(A)];
    B:=iquo(B,2);
  end do:
  A;
```

$$[1, 0, 1, 0, 0, 0, 0, 1]$$

課題：8 桁の整数のそれぞれの桁の値を配列に格納せよ． 8 桁の整数は以下のようにして作られる．

```
> n:=8;
  roll:=rand(10^(n-1)..10^n):
  B:=roll();
```

$$8$$
$$17914675$$

1.2.8 小数点以下 8 桁のそれぞれの桁の数を配列に格納せよ

```
> n:=8:
  roll:=rand(10^(n-1)..10^n):
  B:=evalf(roll()/10^n);
```

$$0.6308447100$$

```
> B:=10*B:
  A:=[]:
  for i from 1 to n do
    A:=[op(A),floor(B)];
    B:=(B-A[i])*10;
  end do:
  A;
```

$$[6, 3, 0, 8, 4, 4, 7, 1]$$

1.2.9 最大数

```
> roll:=rand(1..100):
  n:=5:
  A:=[seq(roll(),i=1..n)];
  i_max:=A[1]:
  for i from 2 to n do
    if (A[i]>i_max) then
      i_max:=A[i];
    end if;
  end do:
  i_max;
```

$$64$$

課題：最小値を求めよ．

1.2.10　ある値の上下で分けた個数

```
> roll:=rand(1..100):
  n:=5: A:=[seq(roll(),i=1..n)];
  i_div:=50:i_low:=0:i_high:=0:
  for i from 1 to n do
    if (A[i]>i_div) then
      i_high:=i_high+1;
    else
      i_low:=i_low+1;
    end if
  end do;
  print(i_low,i_high);
```

```
                                    2, 3
```

1.2.11　素数かどうかの判定

```
> n:=10;
  for i from 1 to n do
    if (isprime(i)) then
      print(i);
    end if;
  end do;
```

1.2.12　2 つの要素の入れ替え

```
> roll:=rand(1..100): n:=5: A:=[seq(roll(),i=1..n)]; sel:=rand(1..n):
  isel:=sel();
  jsel:=sel();
  a:=A[isel]; b:=A[jsel]; A[isel]:=b; A[jsel]:=a;
  A;
```

```
                    [60, 93, 14, 50, 47]
                                2
                                4
                                93
                                50
                                50
                                93
                    [60, 50, 14, 93, 47]
```

より短くするには,

```
> roll:=rand(1..100):
  n:=5:
  A:=[seq(roll(),i=1..n)];
  sel:=rand(1..n):
```

```
isel:=sel();
jsel:=sel();
a:=A[isel];
A[isel]:=A[jsel];
A[jsel]:=a;
A;
```

```
[9, 77, 59, 16, 1]
5
4
1
16
1
[9, 77, 59, 1, 16]
```

1.2.13 コインの表向きの枚数

```
> roll:=rand(0..1):
  n:=10:
  up:=0:
  for i from 1 to n do
    trial:=roll();
    if (trial=1) then
      up:=up+1;
    end if;
  end do:
  up;
```

```
5
```

課題：1..6 のサイコロを 20 回振って，出た目を記録せよ． 記録には，要素が 0 の配列を最初に用意し，出た目を示数にして配列の要素をひとつずつ増やす．

1.2.14 2 次元配列

2 次元配列に対しても同様の操作ができる．ここでは列に対する規格化を示す．

```
> roll:=rand(1..5):
  n:=3:
  A:=[seq([seq(roll(),i=1..n)],j=1..n)];
```

```
A := [[5, 2, 2], [2, 3, 2], [4, 2, 1]]
```

```
> roll:=rand(1..5):
  n:=3:
  A:=[seq([seq(roll(),i=1..n)],j=1..n)];
```

```
1, 1, 5
1, 2, 2
1, 3, 2
2, 1, 2
2, 2, 3
2, 3, 2
3, 1, 4
3, 2, 2
3, 3, 1
```

i,j の順序に注意.

```
> for j from 1 to n do
    tmp:=0;
    for i from 1 to n do
      tmp:=tmp+A[i,j];
    end do;
    for i from 1 to n do
      A[i,j]:=A[i,j]/tmp;
    end do;
  end do:
  A;
```

```
[[5/11, 2/7, 2/5], [2/11, 3/7, 2/5], [4/11, 2/7, 1/5]]
```