Programming—配列(list)—

list

Mapleにはいくつかの配列構造が用意されている．もっとも，頻繁に使うlistを示す．

リスト構造は，中に入れる要素を[]でくくる．

```
> list1:=[1,3,5,7];
```

$$list1 := [1, 3, 5, 7] \tag{1.1.1}$$

要素にアクセスするには，以下のようにインデックスを指定する．

```
> list1[2]; list1[-1]; list1[2..4];
```

$$3$$

$$7$$

$$[3, 5, 7] \tag{1.1.2}$$

-1,-2等は後ろから1番目，2番目を指す．C言語と違い0番目はない．

```
> list1[-1];list1[0];
```

$$7$$

Error, invalid subscript selector

ひとつの要素を書き換えるには，以下のようにする．

```
> list1[3]:=x: list1;
```

$$[1, 3, x, 7] \tag{1.1.3}$$

要素の数，および要素の中身を取り出すには以下のようにする．

```
> nops(list1);
  op(list1);
```

$$4$$

$$1, 3, x, 7 \tag{1.1.4}$$

listへの要素の付け足し

opを用いると，リストに新たな要素を前後，あるいは途中に付け足すことができる．

```
> list1:=[op(list1),9];
```

$$list1 := [1, 3, x, 7, 9] \tag{1.2.1}$$

listlist

listを2次元の配列にするには，括弧2つでくくり，リストリストにする．

```
> l2:=[[1,2,3,4],[1,3,5,7]];
```

$$l2 := [[1, 2, 3, 4], [1, 3, 5, 7]] \tag{1.3.1}$$

要素へのアクセスは以下の通り．

```
> l2[2];
  l2[2][3];
```

$$[1, 3, 5, 7]$$

$$5 \tag{1.3.2}$$

convert, set

ArrayやVectorへの変換はconvertによっておこなう．(出力をメモれ)

```
> convert(l2,Array):
  convert(list1,Vector):
```

[]の代わりに{}(波括弧)で要素をくくると，集合を表わすsetになる．要素の重複を許さず，順番がない．

```
> s1:={1,2,3,3,2};
```

$$s1 := \{1, 2, 3\} \tag{1.4.1}$$

例題

[1,3,5,7]を要素に持つリストを，for-loopとリストへの要素の付け足しを用いて作成せよ．

```
> l1:=[]:
  for i from 1 to 7 by 2 do
    l1:=[op(l1),i];
  end do:
  print(l1);
```

l1の平均を求めよ．

```
> total:=0:
  n:=nops(l1):
  for i from 1 to n do
    total:=total+l1[i];
  end do:
  evalf(total/n);
```

演習

1 iが素数かどうかはisprime(i);で調べることができる．1から100までの素数を要素に持つリストを作れ．