

# Structures Maple

## 1 Les séquences

### 1.1 Définition

Une séquence est une succession d'objets Maple ordonnés séparés par des virgules. Une séquence est un objet de type `exprseq`. Par exemple,

```
> A:=a,2,maple,34,Pi/4; whattype(A);
```

$$A := a, 2, \text{maple}, 34, \frac{\pi}{4}$$

*exprseq*

Pour définir la séquence vide, il faut lui affecter la valeur `NULL` :

```
> B:=NULL;
```

$$B :=$$

Il est possible de concaténer des séquences :

```
> C:=1,2,a,7,f,0;
```

$$C := 1, 2, a, 7, f, 0$$

```
> E:=A,C;
```

$$E := a, 2, \text{maple}, 34, \frac{\pi}{4}, 1, 2, a, 7, f, 0$$

Il existe des méthodes plus élaborées pour définir des séquences. On peut utiliser la fonction `seq` ou l'opérateur `$` :

```
> F:=seq(i^2, i = 1..5);
```

$$F := 1, 4, 9, 16, 25$$

```
> G:=ithprime(i)$i=1..6;
```

$$G := 2, 3, 5, 7, 11, 13$$

Remarquons que la variable  $i$  n'est pas affectée après l'une de ces deux opérations. Attention tout de même, si l'indice qui sert à créer la séquence est déjà affecté dans la feuille de calcul, Maple renvoie un message d'erreur. Voici d'autres exemples d'utilisation de l'opérateur `$` :

```
> x$3;
```

$$x, x, x$$

```
> diff(ln(x), x$3);
```

$$2 \frac{1}{x^3}$$

```
> $3..8;
```

$$3, 4, 5, 6, 7, 8$$

### 1.2 Opérations sur les séquences

Il est possible d'extraire un ou plusieurs éléments d'une séquence comme suit :

```
> A[3]; F[2..4];
```

$$\text{maple}$$

$$4, 9, 16$$

On obtient respectivement le maximum et le minimum d'une séquence à l'aide des fonctions `max` et `min`. On peut aussi utiliser la fonction `sum` (respectivement `product`) pour sommer (resp. multiplier) les éléments d'une séquence :

```
> sum(F[i], i=1..5); product(F[i], i=1..5);
```

$$55$$

$$14400$$

## 2 Les listes

### 2.1 Définition

Les listes sont en fait des séquences entre crochets. Si  $s$  est une séquence, alors  $[s]$  est la liste formée des éléments de  $s$ , dans laquelle ils apparaissent dans le même ordre que dans  $s$ . Une liste peut donc contenir plusieurs fois le même élément. Une liste est du type `list`.

```
> L:=[3,4,5,8,a,b]; whattype(L);
```

$$L := [3, 4, 5, 8, a, b]$$

*list*

On définit la liste vide en ne mettant rien entre les crochets :

```
> M:=[];
```

$$M:=[]$$

### 2.2 Opérations sur les listes

La fonction `nops` permet de calculer la longueur d'une liste, autrement dit le nombre d'éléments que comporte une liste :

```
> nops(L);
```

$$6$$

On peut extraire de deux façons différentes un ou plusieurs éléments d'une liste :

```
> op(4,L); L[5];
```

$$8$$

$$a$$

```
> op(2..5,L); L[2..5];
```

$$[4, 5, 8, a]$$

$$[4, 5, 8, a]$$

Cette extraction peut aussi se faire suivant certains critères grâce à la fonction `select`. Par exemple :

```
> select(x->x>0, [1,-2,3,6,-1]);
```

$$[1,3,6]$$

De même, on peut supprimer les éléments d'une liste qui ne répondent pas à certains critères à l'aide de la fonction `remove` :

```
> remove(x->x<0, [1,-2,3,6,-1]);
```

$$[1, 3, 6]$$

La fonction `map` permet d'appliquer une fonction à tous les éléments d'une liste :

```
> map(ln, [1,2,exp(1)])
```

$$[0, \ln(2), 1]$$

Il existe plusieurs méthodes pour substituer un élément à un autre élément dans une liste. La première méthode consiste à utiliser la fonction `subs` :

```
> subs(5=f, L);
```

$$[3, 4, f, 8, a, b]$$

Notons que cela ne modifie pas la liste d'origine, mais en crée une nouvelle incluant le changement. On peut aussi réaffecter l'élément que l'on veut modifier, mais dans ce cas c'est bien la liste d'origine qui est modifiée :

```
> L[3]:=f; L;
```

$$L_3 := f$$

$$[3, 4, f, 8, a, b]$$

Il est aussi possible de substituer le  $k$ -ième élément d'une liste à l'aide de la fonction `subseq`. Comme pour la fonction `subs`, la liste de départ n'est pas modifiée et Maple crée une nouvelle liste. Par exemple :

```
> subsop(6=maple,L);
```

```
[3, 4, f, 8, a, maple]
```

Pour supprimer un élément, il faut lui substituer la valeur *NULL* :

```
> subs(a=NULL,L);
```

```
[3, 4, f, 8, b]
```

Pour vérifier l'appartenance d'un élément à une liste, on utilise la fonction **member** :

```
> member(3,L);
```

```
true
```

Pour convertir une liste en une séquence,

```
> S:=op(L); whattype(S);
```

```
S := 3, 4, f, 8, a, b
      exprseq
```

Pour concaténer deux listes, on ne peut pas se contenter de les séparer par une virgule comme pour les séquences, il faut d'abord extraire les séquences associées :

```
> N:=[op([1,2,8]),op([9,3])];
```

```
N := [1, 2, 8, 9, 3]
```

La fonction **sort** permet de trier les éléments d'une liste selon un ordre croissant et alphabétique :

```
> sort([d,7,a,A,6,2,F,b])
```

```
[2, 6, 7, A, F, a, b, d]
```

Enfin, comme pour les séquences, on peut sommer et multiplier les éléments d'une liste à l'aide des fonctions **sum** et **product** :

```
> sum(N[i],i=1..nops(N)); product(N[i],i=1..nops(N))
```

```
23
432
```

## 3 Les ensembles

### 3.1 Définition

Les ensembles sont des structures assez proches des listes, à la différence près que les ensembles (comme en mathématiques) ne sont pas ordonnés. On forme un ensemble en ajoutant une paire d'accolades  $\{ \}$  autour d'une séquence. Un ensemble est du type **set**.

```
> P:={1,6,6,3,a,a,a,9}; whattype(P);
```

```
P := {1, 3, 6, 9, a}
      set
```

Remarquons sur l'exemple précédent qu'un ensemble Maple ne peut contenir plusieurs fois le même élément. Comme pour les listes, il est parfois pratique de générer des ensembles en se servant des séquences :

```
> Q:={binomial(6,i)$i=0..6}
```

```
Q := {1, 6, 15, 20}
```

L'ensemble vide se définit simplement en ne mettant rien entre les accolades :

```
> R:={}
```

```
R := {}
```

### 3.2 Opérations sur les ensembles

Les fonctions **op**, **nops** et tout ce que l'on a vu pour extraire un élément d'une liste fonctionnent encore ici. Il existe quelques fonctions supplémentaires qui ne sont utilisables que sur les ensembles. On peut réaliser l'union, l'intersection ou la différence de deux ensembles avec les fonctions **union**, **intersect** et **minus**.

```
> P union Q; P intersect Q; P minus Q;
```

```
{1, 3, 6, 9, 15, 20, a}
  {1, 6}
  {3, 9, a}
```

## 4 Conversion de ces structures

On a vu qu'à partir d'une séquence, une méthode pour générer une liste est d'insérer la séquence entre crochets et, pour générer un ensemble, d'insérer la séquence entre accolades. On a aussi vu qu'avec la fonction `op` on passe d'un ensemble ou d'une liste à une séquence. On peut aussi se servir de la fonction `convert` pour convertir une liste en ensemble et réciproquement.

```
> convert(P,list); convert(L,set);
```

$$\begin{aligned} & [1, 3, 6, 9, a] \\ & \{3, 4, 8, a, b, f\} \end{aligned}$$

La fonction `convert` permet aussi de calculer la somme ou le produit des éléments d'une liste :

```
> convert([1,2,a], '+'); convert([1,2,a], '*');
```

$$\begin{aligned} & 3 + a \\ & 2a \end{aligned}$$

**Exercice 1** Écrire une ou des commandes permettant :

1. d'affecter à  $L$  le résultat de la concaténation des deux listes  $L1$  et  $L2$ , où  $L1 = [4, 8, 1, 2]$  et  $L2 = [1, 4, 2]$ ;
2. de modifier la liste  $L$  en lui ajoutant 0 au début et 8 à la fin;
3. de compter les éléments deux à deux distincts de la liste  $L$ ;
4. de modifier la liste  $L$  en échangeant le premier et le dernier élément de cette liste;
5. de déterminer le plus grand élément de la liste  $L$ ;
6. d'obtenir la liste où tous les éléments de la liste  $L$  égaux à son plus grand élément sont remplacés par 0;
7. d'obtenir la liste dont les éléments sont les carrés des éléments de la liste  $L$ ;
8. d'obtenir la liste dont les éléments sont les éléments de rang pair de la liste  $L$ .

**Exercice 2** Écrire une ou des commandes permettant :

1. d'obtenir la liste des nombres premiers inférieurs à 100 (on pourra consulter l'aide sur `isprime`);
2. d'obtenir la séquence des  $\sin(k\pi/4)$ , l'entier  $k$  variant de 1 à 8, puis d'en supprimer les zéros;
3. d'écrire  $X^6 - 1$  sous forme factorisée (essayer ensuite la commande: `factor`);
4. de calculer  $f^{(5)}(x)$ , où  $f$  est la fonction  $x \mapsto xe^{\frac{1}{x}}$ .

**Exercice 3** Soit  $(\mathcal{E})$  l'équation :  $2x^7 - x^6 - 4x^5 - 15x^4 - 14x^3 + 19x^2 + 28x + 30 = 0$ . Écrire une ou des commandes permettant d'obtenir :

1. la séquence des solutions de l'équation  $(\mathcal{E})$  (on pourra consulter l'aide sur `solve`);
2. la liste des solutions qui sont des nombres réels (resp. qui ont une partie imaginaire strictement positive);
3. la somme et le produit des solutions.

**Exercice 4** Écrire une ou des commandes permettant de calculer :

1.  $\sum_{k=1}^n k^3$ ,  $\sum_{k=0}^n e^{ikx}$ ,  $\prod_{k=1}^{50} k^2$ ,  $\sum_{k=0}^{99} i^k$  (où  $i^2 = -1$ );
2. la somme et le produit des racines sixièmes de l'unité dans  $\mathbb{C}$ ;
3.  $\sum_{0 \leq p, q \leq n} \frac{1}{2^{p+q}}$ ,  $\sum_{k=1}^n \binom{n}{k}$ ,  $\sum_{k=0}^n \left(\frac{-1}{3}\right)^k \binom{n}{k}$ .