

## map

Quelques fonctions supplémentaires inspirées de Maple (*map*)

**Calculatrices :** 89 92+ V200

**Niveaux :** Tale S Sup

**Descriptif :** Groupe de fonctions – Implémentation des fonctions *map* et *zip* de Maple

**Auteur :** Claude Morin

**Mots-clefs :** map, zip, Maple

**Date de dernière révision :** Septembre 2002

### Présentation :

Il s'agit d'un groupe de fonctions implémentant les fonctions *map* et *zip* du logiciel de calcul formel Maple :

- *map1* : fonction map pour une fonction à une variable
- *map2* : fonction map pour une fonction à deux variables
- *zip* : variante de *map2*

Rappelons que la fonction  $map(f, l)$  retourne la liste  $l$  après avoir appliqué  $f$  sur chacun de ses éléments.

### Mode d'emploi :

#### **Map d'une fonction à une variable :**

Il suffit de se placer dans le répertoire dans lequel se trouvent les fichiers, puis de taper  $map1(f, l)$  où  $f$  est la fonction (à une variable) à appliquer et  $l$  la liste ou la matrice sur laquelle appliquer  $f$ .

$f$  peut être indiqué sous notation Maple. Par exemple :  $map("x \rightarrow x+3", \{2, 3\})$  retournera  $\{5, 6\}$ .

#### **Map d'une fonction à deux variables :**

Il suffit de se placer dans le répertoire dans lequel se trouvent les fichiers, puis de taper  $map2(f, l, y)$  où  $f$  est la fonction (à deux variables) à appliquer,  $l$  la liste ou la matrice sur laquelle appliquer  $f$  et  $y$  la seconde variable.

Il est également possible d'appeler la variante  $zip(f, l1, l2)$  où  $f$  est la fonction (à deux variables) à appliquer,  $l1$  et  $l2$  deux listes ou matrices.

### Sources :

Map1 (ff, l1)

```

Func
© entrer une fonction d'1 variable (nom ou notation Maple) et une liste
(ou matrice)
Local ii,nn,vv,ee
If part(dim(l1),0)="{ Then © matrice
  colDim(l1)→nn
  mat▶list(l1)→l1
Else
  0→nn
EndIf
inString(ff,"->")→ii
If ii=0 Then
  seq(#ff(l1[ii]),ii,1,dim(l1))→l1
Else
  left(ff,ii-1)→vv
  right(ff,dim(ff)-ii-1)→ee
  expr(ee&"|"&vv&"=tt")→ee
  seq(ee|tt=l1[ii],ii,1,dim(l1))→l1
EndIf
If nn≠0
  list▶mat(l1,nn)
EndFunc

```

**Map2** (ff,l1,yy)

```

Func
© entrer une fonction de 2 variables (nom ou notation Maple),une liste
(ou matrice) et la seconde variable
Local ii,jj,nn,vv,ee,v1,v2
If part(dim(l1),0)="{ Then
  colDim(l1)→nn
  mat▶list(l1)→l1
Else
  0→nn
EndIf
inString(ff,"->")→ii
If ii=0 Then
  seq(#ff(l1[ii],yy),ii,1,dim(l1))→l1
Else
  left(ff,ii-1)→vv
  inString(vv,",")→jj
  mid(vv,2,jj-2)→v1
  mid(vv,jj+1,dim(vv)-jj-1)→v2
  right(ff,dim(ff)-ii-1)→ee
  expr(ee&"|"&v1&"=tt and "&v2&"=uu")→ee
  seq(ee|tt=l1[ii] and uu=yy,ii,1,dim(l1))→l1
EndIf
If nn≠0
  list▶mat(l1,nn)
EndFunc

```

**Zip** (ff,l1,l2)

```

Func
© entrer une fonction de 2 variables(nom ou notation Maple),et 2 listes
ou matrices
Local ee,ii,jj,nn,vv,v1,v2
If part(dim(l1),0)="{ Then

```

```

colDim(l1)→nn
mat▶list(l1)→l1
mat▶list(l2)→l2
Else
  Ø→nn
EndIf
inString(ff,"->")→ii
If ii=Ø Then
  seq(#ff(l1[ii],l2[ii]),ii,1,dim(l1))→l1
Else
  left(ff,ii-1)→vv
  inString(vv,",")→jj
  mid(vv,2,jj-2)→v1
  mid(vv,jj+1,dim(vv)-jj-1)→v2
  right(ff,dim(ff)-ii-1)→ee
  expr(ee&"|"&v1&"=tt and "&v2&"=uu")→ee
  seq(ee|tt=l1[ii] and uu=l2[ii],ii,1,dim(l1))→l1
EndIf
If nn≠Ø
  list▶mat(l1,nn)
EndFunc

```