

Cours Apl 10 : Encore quelques notions fondamentales : Indicer les données, Tableaux généralisés

* **L'indilage** des données est un concept relativement simple. Ceci consiste à indiquer à Apl de quelle partie d'objet on parle.

Par exemple, nous avons déjà vu le take et le drop (↑ et ↓).

Comment faire si nous voulons extraire les 3ème et 4ème éléments d'un vecteur ?

Illustration :

```
Vec ← 10 × ι 10
Vec[3 4]
30 40
```

L'indilage permet également de modifier les valeurs des objets.

Pour ajouter 100 à Vec[4 5 6] nous écrivons :

```
Vec[4 5 6] ← Vec[4 5 6] + 100
```

ou

```
Vec[4 5 6] + ← 100
```

Pour **indicer une matrice**, on précise d'abord les numéros de lignes, puis de colonnes, séparés par des point-virgules.

Par exemple Mat[5;3] désigne la valeur se trouvant en ligne 5, colonne 3 de Mat.

Pour indicer un cube, on mentionne les coordonnées dans l'ordre : plans, lignes, colonnes.

En règle générale l'indilage est toujours effectué de la plus grande à la plus petite dimension.

Revenons aux matrices. Si, par exemple, on veut désigner une colonne, avec toutes les lignes de la matrice, on laisse vide l'emplacement destiné à indiquer les numéros de lignes. Bien évidemment, ceci fonctionne également pour sélectionner des lignes entières.

Exemple :

```
Mat ← 4 4 ρ 'aaaabbbbccccdddd'
```

Afficher les 2 premières colonnes :

```
Mat[;1 2]
```

```
aa
bb
cc
dd
```

Afficher les lignes 2 et 3 :

```
Mat[2 3;]
```

```
bbbb
cccc
```

- On peut également indiquer une matrice avec des couples afin de désigner des données situées à des emplacements non contigus.

Exemple :

```
Mat ← 3 3 ρ ι 9
Mat
1 2 3
4 5 6
7 8 9
```

Pour afficher la diagonale allant d'en haut à gauche à en bas à droite, on demande :

```
Mat[(1 1)(2 2)(3 3)]
1 5 9
```

On peut également écrire :

```
Mat[2ρ ι 1 ρ Mat]
ou
Mat[2ρ ι 1 ρ Mat]
1 5 9
```

Cette ligne de code comporte pas mal de nouveautés.

Reprenons toutefois la précédente.

Si vous demandez : $\rho (1\ 1)(2\ 2)(3\ 3)$

Apl vous répondra 3.

En effet, cet objet est un "vecteur généralisé".

On appelle vecteur généralisé, un vecteur composé d'objets autres que des scalaires.

C'est une structure (vecteur, matrice, ...) composée de structures.

Dans notre exemple, c'est un vecteur composé de trois vecteurs de 2 éléments.

La deuxième manière de composer la diagonale : $2\rho \iota 1 \rho Mat$ comporte notamment l'expression : $1 \rho Mat$

ρ **est appelée Disclose**. Cette fonction extrait un élément d'un vecteur. La position de l'élément à extraire est à gauche et le vecteur, à droite.

Ainsi, $1 \rho Mat$ rend 3.

$\iota 3$ rend 1 2 3

Venons en au **each**. C'est un opérateur, un peu comme la réduction. Il signifie que la fonction immédiatement à sa gauche va s'appliquer à chacun des éléments à sa droite.

Par exemple : 1 2 3 , 4 donne

```
1 2 3 4
```

Alors que 1 2 3 , each 4 donne

```
1 4 2 4 3 4
```

Dans le premier cas, on a simplement concaténé 4 à la suite du vecteur 1 2 3.

Avec le each, on a collé un 4 à chaque élément du vecteur 1 2 3.

Pour en revenir à l'exemple initial :

$2\rho \text{ `` } 1\ 2\ 3$ donne donc :

$1\ 1\ 2\ 2\ 3\ 3$ qu'on peut écrire également $(1\ 1)\ (2\ 2)\ (3\ 3)$

puisqu'on fabrique un vecteur de 2 éléments à partir de chaque élément de $1\ 2\ 3$.

De même, pour connaître la taille des vecteurs résultant de $2\rho \text{ `` } 1\ 2\ 3$, il suffit de demander :

$\rho \text{ `` } 2\rho \text{ `` } 1\ 2\ 3$
 $2\ 2\ 2$

Avantage de cette façon de faire : sans connaître la taille d'une matrice carrée, vous en afficherez toujours la diagonale ($2\rho \text{ `` } 1 \rightarrow \rho \text{ Mat}$).

Autre exemple :

$VV \leftarrow 'titi'\ 'tata'\ 'toto'\ 'et\ moi'$

Pour ajouter un 'k' derrière chaque élément, il suffit d'écrire :

$VV, \text{ `` } 'k'$
 $titik\ tatak\ totok\ et\ moik$

'k' étant un scalaire, il s'est propagé sur chaque élément de VV .

Que donne ceci : $VV, \text{ `` } 'abcd'$

$titia\ tatab\ totoc\ et\ moid$

Ici le each a pris chaque élément de $abcd$ et l'a collé à l'élément de VV correspondant.

Comment coller 'plouf' à chaque élément de VV ?

$VV, \text{ `` } 'plouf'$

Lenght Error

Apl refuse. En effet, il a 2 possibilités, soit il propage un scalaire, soit il combine les 2 objets terme à terme, à condition qu'ils aient la même taille, ce qui n'est pas le cas, d'où un message d'erreur de longueur.

Dans notre cas, nous voudrions que 'plouf' se comporte comme un scalaire et se colle à chaque élément de VV .

Il existe pour cela la fonction contraire de disclose : **enclose (c)**.

Elle transforme en scalaire généralisé l'objet auquel elle est appliquée.

Illustration :

$\rho\ 'plouf'$

6

$\rho\ c\ 'plouf'$

rien

$\rho\ \text{ `` } c\ 'plouf'$

6

Donc, pour en revenir à notre problème de concaténation, la solution est :

$VV, \text{ `` } c\ 'plouf'$

$titiplouf\ tataplouf\ totoplouf\ et\ moiplouf$

Parmi les workspaces fournis avec Dyalog Apl, figure "util".
Comme son nom l'indique il contient un certain nombre de fonctions utilitaires, dont la fonction DISPLAY.

Recopiez-la dans votre workspace en tapant :
)COPY util DISPLAY

Maintenant, essayez ceci :

```
VV ← 'titi' 'tata' 'toto' 'et moi'
```

puis

```
DISPLAY 2↑VV, '' c ' plouf'
```

Apl devrait vous répondre ceci :

```
|→-----|
| |→-----| |→-----| |
| |titi plouf| |tata plouf| |
| |-----| |-----| |
|ε-----|
```

L'outil DISPLAY permet d'obtenir une représentation visuelle des structures des objets. On voit ici un vecteur comportant 2 autres vecteurs.

- Take et drop monadiques (↑ et ↓) :

Ces deux fonctions font le contraire l'une de l'autre. Elles travaillent par défaut sur la dernière dimension.

Si on applique ↓ à une matrice, on obtient un vecteur de vecteurs constitués par les lignes de la matrice.

Ainsi, ↓Mnum1 rend :

```
10 20 30 40 1 2 3 4 1 1 1 1
```

Visualisons ce résultat avec DISPLAY :

```
DISPLAY ↓Mnum1
```

```
|→-----|
| |→-----| |→-----| |→-----| |
| |10 20 30 40| |1 2 3 4| |1 1 1 1| |
| |~-----| |~-----| |~-----| |
|ε-----|
```

Pour obtenir un vecteur de vecteur de colonnes, il faut écrire :

```
↓[1]Mnum1
```

La fonction drop a donc pour effet de transformer une dimension en profondeur.

A l'inverse, la fonction take transforme une profondeur en dimension.

Par exemple, ↑VV restituera une matrice dont chaque ligne sera un des éléments du vecteur de vecteurs initial.

```
↑VV rend
```

```
titi
tata
toto
et moi
```

Travaux pratiques :

0. Chargez votre Ws de travaux pratiques :

```
)load c:\Mes documents\pratique-apl
```

Créez la matrice du personnel de votre PME :

```
Perso←4 5ρ 'Pierre' 'Sylvie' 'Paul' 'Jacques' 'Lea' 'H' 'F' 'H' 'H' 'F' 25 27 30 28 22 40 45 43 50 48
```

Cette matrice contient en 1ère ligne les noms des personnes, en deuxième leur sexe, en troisième leur age et en quatrième leur salaire brut annuel.

1. Affichez la avec la fonction DISPLAY

2. Affichez les noms et salaires des personnes gagnant plus de 45 KEuros par an

3. Affichez :

- le nom, l'age et le salaire de la femme la moins payée
- le nom, l'age et le salaire de la femme la mieux payée

4. Quel est le salaire moyen :

- de l'entreprise
- des hommes
- des femmes
- des 20-26 ans
- des plus de 26 ans
- si on prend le salaire moyen des hommes comme référence, quel est le % d'écart avec celui des femmes.

5. Combien y a-t-il d'hommes dans l'entreprise ?

De femmes ?

Affichez une matrice Stats de 4 lignes x 2 colonnes avec

- en première ligne un H et un F
- en deuxième, le nombre de personnes du sexe correspondant
- en troisième son % en effectif
- en quatrième le salaire moyen par sexe

6. **La fonction \uparrow rend le vecteur de tri croissant d'un objet.**

Par exemple \uparrow 'cdba' rend 4 3 1 2

et 'cdba'[4 3 1 2] donne bien 'abcd'

Sachant cela :

- affichez la matrice des noms et ages par ordre croissant d'age
- affichez la matrice des noms et salaires par ordre décroissant de salaire (fonction \downarrow).
- affichez Perso triée par ordre alphabétique.

Attention ! Le tri ne fonctionne pas sur un vecteur de vecteurs. En revanche, il accepte tout à fait les simples matrices de texte.

7. Affichez la première lettre de chaque prénom

- Affichez le nombre de caractères de chaque prénom.
- Affichez tous les prénoms comportant au moins un "a".
- Affichez tous les prénoms commençant par un "P".

8. Quelle est la masse salariale de l'entreprise ?

- Quelle serait la masse salariale de l'entreprise si les salaires étaient plafonnés à 47 ?
- Quels seraient les salaires de chacun (nom, ancien salaire, augmentation et nouveau salaire) si tout le monde était augmenté de 10% dans la limite de 4.3 d'augmentation ?
- Quel serait le pourcentage d'augmentation de la masse salariale ?

9. Votre activité marche bien. Vous embauchez Bob, un brillant styliste américain de 23 ans qui débute chez vous à 47 KEuros.

- Ajoutez le donc dans Perso.
- Refaites les stats des points 4 et 8.

10. Finalement, Bob n'a même pas mis les pieds chez vous et est allé chez un de vos concurrents. Qu'à cela ne tienne, Lucia, une charmante italienne de 24 ans était très intéressée par le poste et c'est "molto contenta" qu'elle commence sa carrière chez vous pour 45 KEuros.

- Ajoutez la dans Perso à la place de ce petit prétentieux de Bob.
- Refaites les stats des points 4 et 8.

11. Sauvez votre travail :

)save

Solutions :

Créez la matrice du personnel de votre PME :

```
Perso←4 5ρ'Pierre' 'Sylvie' 'Paul' 'Jacques' 'Lea' 'H' 'F' 'H' 'H' 'F' 25 27  
30 28 22 40 45 43 50 48
```

1. Affichez la avec la fonction *DISPLAY*

```
DISPLAY Perso
```

2. Affichez les noms et salaires des personnes gagnant plus de 45 KEuros par an

```
(Perso[4;]>45)/Perso[1 4;]
```

3. Affichez :

- le nom, l'age et le salaire de la femme la moins payée

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

Décomposons cette grande ligne :

- $Perso[2;]='F'$ nous donne le booléen des femmes

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

- $(Perso[2;]='F')/Perso[4;]$ nous donne les salaires des femmes et, en ajoutant $\backslash/$ devant, on trouve le plus petit salaire féminin.

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

- $Perso[4;]=\/(Perso[2;]='F')/Perso[4;]$ nous indique parmi tous les salaires de l'entreprise, lesquels sont égaux au plus petit salaire féminin.

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

Ensuite on cherche parmi les femmes ($Perso[2;]='F'$), la personne qui a un salaire égal au plus petit salaire féminin $Perso[4;]=\/(Perso[2;]='F')/Perso[4;]$:

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

On applique ensuite ce booléen aux lignes 1, 3 et 4.

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

- le nom, l'age et le salaire de la femme la mieux payée

```
((Perso[2;]='F')^Perso[4;]=\/(Perso[2;]='F')/Perso[4;])/Perso[1 3 4;]
```

4. Quel est le salaire moyen :

- de l'entreprise

$$(+/Perso[4;])\div\rho Perso[4;]$$

- des hommes

$$(+/(Perso[2;]='H')/Perso[4;])\div+/Perso[2;]='H'$$

Décomposons :

$+/Perso[2;]='H'$ nous donne le nombre d'hommes

$$(+/(Perso[2;]='H')/Perso[4;])\div+/Perso[2;]='H'$$

$+(Perso[2;]='H')/Perso[4;]$ représente la somme des salaires masculins

$$(+/(Perso[2;]='H')/Perso[4;])\div+/Perso[2;]='H'$$

Pour obtenir la moyenne on divise donc cette somme par le nombre de personnes.

$$(+/(Perso[2;]='H')/Perso[4;])\div+/Perso[2;]='H'$$

Pour alléger l'écriture on peut procéder en 2 temps :

- Créer un booléen

- l'utiliser dans le calcul

$$Bool\leftarrow Perso[2;]='H'$$
$$(+/Bool/Perso[4;])\div+/Bool$$

- *salaire moyen des femmes*

$$(+/(Perso[2;]='F')/Perso[4;])\div+/Perso[2;]='F'$$

OU

$$Bool\leftarrow Perso[2;]='F'$$
$$(+/Bool/Perso[4;])\div+/Bool$$

- *des 20-26 ans*

$$Bool\leftarrow (Perso[3;]\geq 20)\wedge Perso[3;]\leq 26$$
$$(+/Bool/Perso[4;])\div+/Bool$$

- *des plus de 26 ans*

$$(+/(Perso[3;]>26)/Perso[4;])\div+/Perso[3;]>26$$

OU

$$Bool\leftarrow Perso[3;]>26$$
$$(+/Bool/Perso[4;])\div+/Bool$$

- si on prend le salaire moyen des hommes comme référence, quel est le % d'écart avec celui des femmes.

On calcule d'abord le salaire moyen des hommes :

```
SH←(+/(Perso[2;]='H')/Perso[4;])÷+ /Perso[2;]='H'
```

Puis celui des femmes :

```
SF←(+/(Perso[2;]='F')/Perso[4;])÷+ /Perso[2;]='F'
```

Et en final le % d'écart :

```
100×(SF-SH)÷SH
```

5. Combien y a-t-il d'hommes dans l'entreprise ?

```
+ /Perso[2;]='H'
```

De femmes ?

```
+ /Perso[2;]='F'
```

Affichez une matrice Stats de 4 lignes x 2 colonnes avec

- en première ligne un H et un F

```
Stats←1 2ρ'H' 'F'
```

- en deuxième, le nombre de personnes du sexe correspondant

```
Stats,[1]←(+ /Perso[2;]='H'),(+ /Perso[2;]='F')
```

- en troisième son % en effectif

```
Stats,[1]←100×Stats[2;]÷+ /Stats[2;]
```

- en quatrième le salaire moyen par sexe

```
Stats,[1]←((+ /Perso[2;]='H')/Perso[4;])÷Stats[2;1],((+ /Perso[2;]='F')/Perso[4;])÷Stats[2;2]
```

Explication :

$(+ /Perso[2;]='H')/Perso[4;]$ donne le salaire total des hommes et on le divise par le nombre d'hommes récupéré dans Stats[2;1]. On fait la même chose pour les femmes.

6. La fonction Δ rend le vecteur de tri croissant d'un objet.

- affichez la matrice des noms et ages par ordre croissant d'age

```
Perso[1 3; ΔPerso[3;]]
```

- affichez la matrice des noms et salaires par ordre décroissant de salaire (fonction Ψ).

```
Perso[1 4; ΨPerso[4;]]
```

- affichez Perso triée par ordre alphabétique.

```
Perso[; Δ↑Perso[1;]]
```

La fonction de tri n'admet pas de vecteur généralisé en argument droit.

Pour obtenir le vecteur de tri de la matrice en fonction des noms, on est donc obligé de transformer le

vecteur de prénoms en une matrice alpha. C'est pour cette raison qu'on écrit $\uparrow Perso[1;]$

7. Affichez la première lettre de chaque prénom

```
1↑Perso[1;]
```

- Affichez le nombre de caractères de chaque prénom.

```
ρPerso[1;]
```

- Affichez tous les prénoms comportant au moins un "a".

```
(v/`Perso[1;]='a')/Perso[1;]
```

- Affichez tous les prénoms commençant par un "P".

A priori, on serait tenté d'écrire :

```
('P'=1↑Perso[1;])/Perso[1;]
```

Malheureusement, dans ce cas, apl répond :

DOMAIN ERROR

Pourquoi ?

Utilisons DISPLAY pour y voir plus clair :

```
DISPLAY 'P'=1↑Perso[1;]
```

```
|→-----|
| |→| |→| |→| |→| |→| |
| |1| |0| |1| |0| |0| |
| |-| |-| |-| |-| |-| |
|ε-----|
```

On voit que ce booléen n'est pas un simple vecteur mais un vecteur de vecteurs de 1 élément chacun qu'Apl refuse d'appliquer à Perso[1;]

En effet on a à gauche des vecteurs de 1 élément qui ne peuvent pas être appliqués aux vecteurs de plus d'un élément de l'objet droite. En revanche, si le vecteur de booléens est un vecteur simple, chacun de ses scalaires sera propagé sur le prénom correspondant.

Il faut donc transformer cet objet en un vecteur simple.

Le take monadique va donc nous faire remonter d'une profondeur mais rendra une matrice, qu'il faudra linéariser.

```
DISPLAY ,↑'P'=1↑Perso[1;]
```

```
|→-----|
|1 0 1 0 0|
|~-----|
```

Cette fois, c'est bien un simple vecteur booléen parfaitement utilisable :

```
(,↑'P'=1↑Perso[1;])/Perso[1;]
```

8. Quelle est la masse salariale de l'entreprise ?

`+ / Perso[4;]`

- Quelle serait la masse salariale de l'entreprise si les salaires étaient plafonnés à 47 ?

`+ / Perso[4;][47]`

- Quels seraient les salaires de chacun (nom, ancien salaire, augmentation et nouveau salaire) si tout le monde était augmenté de 10% dans la limite de 4.3 d'augmentation ?

`Augm←4.3 [Perso[,4;]]×.1
Perso[1 4;],[1]Augm,[1]Perso[,4;]+Augm`

Commentaires :

- on commence par calculer les augmentations plafonnées dans une matrice d'une ligne.

`ρPerso[4;]` vaut 5 alors que `ρPerso[,4;]` vaut 1 5.

Par défaut, si on demande à apl une seule colonne ou ligne d'une matrice, il rend un vecteur. Par contre, si on demande plus d'une ligne et/ou colonne on obtiendra bien une matrice.

Donc, quand on indice avec un scalaire, on obtient un vecteur et quand on indice avec un ou 2 vecteurs, on obtient une matrice.

`,4` est un vecteur de 1 élément.

- Ensuite, on affiche les uns en dessous des autres les éléments désirés.

- Quel serait le pourcentage d'augmentation de la masse salariale ?

`100×(+ / Augm)÷+ / Perso[4;]`

9. Votre activité marche bien. Vous embauchez Bob, un brillant styliste américain de 23 ans qui débute chez vous à 47 KEuros.

- Ajoutez le donc dans Perso.

`Perso,←'Bob' 'H' 23 47`

- Refaites les stats des points 4 et 8.

Vous, pas moi ...

10. Finalement, Bob n'a même pas mis les pieds chez vous et est allé chez un de vos concurrents. Qu'à cela ne tienne, Lucia, une charmante italienne de 24 ans était très intéressée par le poste et c'est "molto contenta" qu'elle commence sa carrière chez vous pour 45 KEuros.

- Ajoutez la dans Perso à la place de ce petit prétentieux de Bob.

`Perso[;←1ρPerso]←'Lucia' 'F' 24 45`

- Refaites les stats des points 4 et 8.

Courage ...