

## La gestion des collections

*NB : Utiliser la feuille de syntaxe*

### 1 Exercice 1 – les tableaux

Ecrivez un programme qui commence par demander le nombre N de notes, et ensuite demande chacune des notes. Les notes seront stockées dans un tableau de taille N  
Ensuite, le programme demande le coefficient de chaque note et stocke aussi ces coefficients dans un tableau.

Votre programme calcule ensuite la moyenne des notes en tenant compte des coefficients et affiche le résultat.

Son exécution se présente ainsi :

```
Merci de saisir le nombre de notes :  
3  
Merci de saisir la note 1  
12,5  
Merci de saisir la note 2  
15  
Merci de saisir la note 3  
10,5  
Merci de saisir le coef 1  
2,5  
Merci de saisir le coef 2  
5  
Merci de saisir le coef 3  
3  
Votre moyenne est 13.119047619047619
```

Vous noterez que la saisie des nombres réels se fait avec une virgule et non un point !! (norme française).

## 2 Exercice 2 – les tableaux

Modifier le programme précédent pour que celui affiche à la fin :

- la note maxi
- la note mini
- la liste des notes triées de la plus grande à la plus petite

Pour le tri, vous utiliserez la méthode du tri à bulles vue en cours.

## 3 Exercice 3 – les arraylists

Réalisez le même programme qu'à l'exercice 1, mais en utilisant maintenant deux ArrayList (un pour les notes, un pour les coefficients). Du point de vue de l'utilisateur, la seule différence est la suivante : il ne donne pas au départ le nombre de notes, mais il saisit -1 pour dire qu'il a donné la dernière note.

Son exécution se présente ainsi :

```
Merci de saisir la note (-1 si vous avez fini)
12,5
Merci de saisir la note (-1 si vous avez fini)
15
Merci de saisir la note (-1 si vous avez fini)
10,5
Merci de saisir la note (-1 si vous avez fini)
-1
Merci de saisir le coef 1
3
Merci de saisir le coef 2
3
Merci de saisir le coef 3
1
Votre moyenne est 13.285714285714286
```

## 4 Exercice 4

Modifier le programme de l'exercice 3 pour que celui affiche à la fin :

- la note maxi
- la note mini
- la liste des notes triées de la plus grande à la plus petite

Pour le tri, vous utiliserez la méthode du tri à bulles vue en cours.

## 5 Exercice 5 – encore des tableaux

Il existe une méthode de détermination de tous les nombres premiers entre 1 et  $n$ , connue sous le nom de crible d'Eratosthène.

Elle consiste à dresser une liste de tous les nombres entiers considérés (ici donc de 1 à  $n$ ), et ensuite à rayer dans la liste tous les multiples d'autres entiers.

Plus précisément, on procède ainsi :

- on raye le 1 (qui par définition n'est pas un nombre premier)
- on prend 2, qui est premier, et on raye tous les multiples de 2 dans la liste
- on prend le nombre suivant 2 qui n'a pas été rayé, qui est donc premier. Celui ci est donc premier, et on raye tous les multiples de ce nombre
- on constitue ainsi de suite, jusqu'à ce que le nombre premier considéré soit supérieur à la racine carrée de  $n$ . On peut alors démontrer que tous les nombres non premiers ont été rayés de la liste.

Ecrire un programme exploitant cette méthode pour rechercher tous les nombres premiers compris entre 1 et une valeur fournie en donnée.

```
Donnez le plus grand nombre entier a examiner : 1000
entre 1 et 1000 les nombres premiers sont :
1 2 3 5 7 11 13 17 19 23
29 31 37 41 43 47 53 59 61 67
71 73 79 83 89 97 101 103 107 109
113 127 131 137 139 149 151 157 163 167
173 179 181 191 193 197 199 211 223 227
229 233 239 241 251 257 263 269 271 277
281 283 293 307 311 313 317 331 337 347
349 353 359 367 373 379 383 389 397 401
409 419 421 431 433 439 443 449 457 461
463 467 479 487 491 499 503 509 521 523
541 547 557 563 569 571 577 587 593 599
601 607 613 617 619 631 641 643 647 653
659 661 673 677 683 691 701 709 719 727
733 739 743 751 757 761 769 773 787 797
809 811 821 823 827 829 839 853 857 859
863 877 881 883 887 907 911 919 929 937
941 947 953 967 971 977 983 991 997
```

## 6 Exercice 6 – les nombres distincts

Ecrivez un programme qui commence par demander le nombre  $N$  de notes, et ensuite demande chacune des notes. Les notes seront stockées dans un tableau de taille  $N$ .

Affichez ensuite le nombre de notes distinctes. Par exemple, si les notes sont 12 13 12 14 12, votre programme doit alors afficher 3 (il y a 3 notes distinctes : 12 13 et 14).

Faites le même exercice avec des ArrayList.

## 7 Exercice Bonus - 7 – la calculatrice.

Ecrivez un programme qui simule une calculatrice.

Votre programme commencera par demander l'opérande 1 , puis l'opérateur (qui pourra être égal à + , - , / , \* et%), puis l'opérande 2, puis affichera le résultat sous la forme :

```
24 + 15 = 39
```

Un message d'erreur clair devra apparaître en cas d'opération interdite (division par 0 par exemple).

Après affichage du résultat, l'opérateur pourra passer à une autre opération.

## 8 Exercice Bonus - 8 – Histogrammes aléatoires

Faire un programme qui demande un nombre N en entrée.

Faire un programme qui calcule N nombres aléatoires compris entre 0 et 100 et qui les stocke dans un tableau.

Pour calculer un nombre aléatoire compris entre 0 et 99, vous utiliserez la formule suivante

```
int a = (int) (Math.random()*101) ;
```

Tracez ensuite un histogramme de la répartition de ces nombres à l'écran. Vous obtiendrez quelque chose comme ceci :

15%	*****			
..	*****			
10%	*****			
..	*****			
05%	*****			
04%	*****			*****
03%	****	*****		*****
02%	****	*****	*****	*****
01%	****	*****	*****	*****
00%	****	*****	*****	*****
	0-10	11-20	21-30	.... 90-100

Dans cet exemple , les nombres entre 0 et 10 représentent 3 % du total, les nombres entre 11 et 20 représentent 15% du total, les nombres entre 90 et 100 représentent 4 % du total.

Que se passe t il quand N devient grand ?

## 9 Exercice Bonus - 9 – Ecrire un chèque

Faites un programme capable d'écrire un chèque entre 1 et 10 000 euros : vous entrez par exemple 951 , votre programme vous écrit alors en toutes lettres

```
neuf cent cinquante et un euros
```