Département d'Informatique
DUT Informatique S1
Année 2014/2015
Algorithmique et programmation

Algorithmique et programmation
TP 11 : Le langage Java, les tableaux

1 Tableaux

Cet exercice a pour seul but de vérifier que vous savez déclarer, construire et utiliser les tableaux en Java.

Écrire un petit programme qui déclare un tableau d'entiers de taille prédéfinie puis qui, à l'aide d'un menu interactif, permet d'effectuer les opérations suivantes :

- afficher la taille du tableau ;
- afficher le contenu du tableau
- modifier l'élément à l'indice i ;
- échanger les valeurs de deux éléments ;
- compter le nombre d'occurrences d'une valeur ;
- ...

2 Triangle de Pascal

Écrire un programme qui affiche le triangle de Pascal jusqu'au rang N, donné par l'utilisateur. Pour le calcul de la ligne courante (n), on s'appuiera sur la ligne précédente :

$$\begin{cases} cour[0] = 1 \\ cour[i] = prec[i-1] + prec[i], \quad \text{pour tout } i \in [1; n-1] \\ cour[n] = 1 \end{cases}$$

Il est demandé d'utiliser un tableau pour calculer une ligne entière, avant le l'afficher. Pour cela, vous déclarerez vos tableau avec une taille suffisamment grande, quitte à ne pas utiliser le tableau dans son intégralité.

Exemple d'exécution :

```
N? 6
[0] -- 1
[1] -- 1 1
[2] -- 1 2 1
[3] -- 1 3 3 1
[4] -- 1 4 6 4 1
[5] -- 1 5 10 10 5 1
[6] -- 1 6 15 20 15 6 1
```

3 Occurrences

Écrire un programme qui lit un texte sur l'entrée standard, compte le nombre d'occurrences de chaque lettre minuscule (`'a'`... `'z'`), puis affiche les résultats.

Compléter ensuite le programme pour qu'il compte aussi les lettres majuscules. Le faire afficher les résultats sous forme d'histogramme horizontal (facile) ou vertical (plus difficile).

4 Somme préfixe

a. Écrire une fonction Java `sommePrefixe()` prenant en paramètres un entier N et deux tableaux `in` et `out` de dimension N, et calculant les valeurs de `out` suivant la formule suivante ($i \in [0..N-1]$) :

$$\texttt{out}[i] = \texttt{in}[0] + \texttt{in}[1] + \cdots + \texttt{in}[i]$$

Exemple. Soit $N = 10$, si le tableau `in` contient les valeurs suivantes :

1	2	3	4	5	6	7	8	9	10

alors, après l'appel de fonction `sommePrefixe(10, in, out)`, le tableau `out` contient les valeurs suivantes :

1	3	6	10	15	21	28	36	45	55

b. Écrire ensuite un programme Java qui :

(i) demande 10 nombres à l'utilisateur ;

(ii) calcule la somme préfixe du tableau formé par ces 10 nombres ;

(iii) affiche le résultat.

5 Émile

Émile est fier de son nouveau cadenas et de la combinaison à six chiffres qu'il a trouvée. Il se permet même de donner les indications suivantes à José :

« Supposons que l'on écrive une liste d'entiers dans l'ordre naturel 1,2,3,4,5... Le premier chiffre de la combinaison est le chiffre qui vient, dans l'ordre d'écriture, juste après le millième chiffre 1 écrit ; le second chiffre de la combinaison est celui qui vient après le 2000^{e} chiffre 1, et ainsi de suite jusqu'au 6^{e} chiffre qui est, bien sûr, celui qui suit immédiatement le 6000^{e} chiffre 1 ! ».

Émile ne se doute pas que José trouvera la combinaison !

Et vous, êtes-vous capable d'écrire un algorithme (puis un programme) qui puisse retrouver cette combinaison ?

Vous écrirez une fonction qui recherche la combinaison et la stocke dans un tableau passé en paramètre. Le programme principal utilisera cette fonction pour trouver la combinaison, avant de l'afficher [1].

1. Sauf erreur, la combinaison est 4-6-6-9-0-1.