

## TD 10 Révisions

**Exercice 1 : Approximation du nombre d'Euler.** Le nombre d'Euler  $e$  est égal à  $\sum_{k=0}^{\infty} \frac{1}{k!}$ . Donner une approximation de  $e$  en calculant la somme des  $n$  premiers termes de la suite  $(\frac{1}{k!})_{k \geq 0}$ .

**Exercice 2 : Eratosthène.** Il existe une méthode de détermination de tous les nombres premiers compris entre 1 et  $n$ , connue sous le nom de *crible d'Eratosthène*. Elle consiste à dresser une liste de tous les nombres entiers considérés et à y *raier* 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 recherche, à partir du dernier nombre premier considéré (la première fois, on convient qu'il s'agit du 1), le premier nombre non rayé (on peut montrer qu'il est premier). Il devient, à son tour, le dernier nombre premier considéré et on raye tous ses multiples,
- on répète le traitement précédent 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. Écrire un programme exploitant cette méthode pour rechercher tous les nombres premiers compris entre 1 et une valeur fournie en donnée.

**Exercice 3 : Récursivité.** Ecrire une fonction calculant la valeur de la *fonction d'Ackermann*  $A$  définie pour  $m \geq 0$  et  $n \geq 0$  par :

- $A(m, n) = A(m - 1, A(m, n - 1))$  pour  $m > 0$  et  $n > 0$
- $A(0, n) = n + 1$  pour  $n > 0$ ,
- $A(m, 0) = A(m - 1, 1)$  pour  $m > 0$ .

**Exercice 4 : Les mots de Fibonacci.** La suite  $(S_n)$  des mots de Fibonacci est définie par  $S_1 = 1$ ,  $S_2 = 0$ , et pour tout  $n > 2$ , le mot  $S_n$  par la concaténation des mots  $S_{n-1}$  et  $S_{n-2}$  (dans cet ordre). Faire un programme Java permettant d'afficher  $S_n$  lorsque  $n$  est saisi par l'utilisateur. Refaire ce programme en utilisant la récursivité.

**Exercice 5 : Conjugaison.** Ecrire un programme qui lit au clavier un verbe du premier groupe (il s'assurera qu'il est bien terminé par *er*) et qui en affiche la conjugaison au présent de l'indicatif. On supposera qu'il s'agit d'un verbe régulier. Autrement dit, on admettra que l'utilisateur ne fournit pas un verbe tel que manger (dans ce cas, le programme affichera nous mangons!). Les résultats se présenteront ainsi :

Donnez un verbe regulier du premier groupe :

dire

\*\*\* il ne se termine pas par er - donnez-en un autre :

chanter

je chante

tu chantes

il/elle chante

nous chantons

vous chantez

ils/elles chantent

**Exercice 6 : Tri de mots.** Ecrire un programme qui lit une suite de mots au clavier et qui les affiche triés par ordre alphabétique. On supposera que ces mots ne contiennent que des lettres non accentuées (majuscules ou minuscules). On ne saisira pas plus de 10 mots.