
TP1 Java : Pour bien commencer

Le travail est individuel. Les programmes informatiques sont à envoyer sous forme d'une archive .tar ou .zip par email à `petru.valicov@ens-lyon.fr`. L'archive doit contenir le nom et le prénom. La date limite est fixée au dimanche 17/11/2013.

Avant-propos : tout programme informatique doit être testé! Et n'oubliez pas que
Program testing can be used to show the presence of bugs, but never to show their absence!

Edsger W. Dijkstra

1 Échauffement

Échauffement 1 (min) Écrire un programme Java qui saisit au clavier trois entiers et affiche le plus petit d'entre eux.

Échauffement 2 (swap)

1. Écrire un programme Java qui échange les valeurs des deux variables saisies au clavier et qui affiche les nouvelles valeurs (j'ai un pot de bière et un pot de vin, comment je fais ???).
2. Imaginez-vous être au début des années '50. C'est les années d'après-guerre, la mémoire vive coûte très cher... Échangez deux variables sans utiliser de variable auxiliaire.

C'est fini toutes ces lignes de code dans le méthode `main`! Désormais on va écrire que des fonctions qu'on pourra éventuellement réutiliser comme briques de base.

Échauffement 3 (permutations) Écrire une fonction qui saisit au clavier un entier n et affiche la valeur de $n!$. Pensez à écrire deux version : en récursif et en itératif.

Échauffement 4 (somme) Écrire une fonction **naïve** prenant en entrée un entier n et retournant la somme des carrés des nombres entiers de 1 à n . Tester sur des exemples. On pourra vérifier la correction du programme en utilisant la formule :

$$\sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}.$$

Échauffement 5 (Fibonacci) Écrire une fonction qui retourne le $n^{\text{ème}}$ terme de la *suite de Fibonacci* définie par :

$$\begin{cases} u(0) = 0 \\ u(1) = 1 \\ u(n+2) = u(n+1) + u(n) \text{ pour } n \geq 0 \end{cases}$$

Écrire un programme qui demande deux entiers n_1 et n_2 positifs et qui affiche la somme $u(n_1) + u(n_2)$.

Échauffement 6 (les premiers ne sont pas drôles...) Écrire une fonction prenant en entrée un entier n et affichant un message indiquant si n est un nombre premier ou non. Tester sur des exemples.

2 Les tableaux

Ératosthène (...mais vraiment pas drôles!)

On désigne sous le nom de *crible d'Ératosthène* une méthode de recherche des nombres premiers plus petits qu'un entier n donné. Pour ceci, on écrit la liste de tous les entiers de 2 à n , ensuite on répète les opérations suivantes jusqu'à ce qu'on atteigne la racine carrée de n :

- on considère le plus petit nombre non-souligné
- on souligne tous les multiples non-souignés de ce nombre

À l'issue de cet algorithme, les entiers non-souignés sont premiers.

En utilisant un tableau de booléen pour représenter la liste des entiers, écrire un programme qui utilise des sous-programmes pour implanter cette méthode de recherche des nombres premiers.

Pascal (je croyais qu'on codait en Java :-/)

Écrire une fonction qui prend en entrée un nombre n et qui construit les n premières lignes du triangle de Pascal. Exemple pour 5 premières lignes :

$$\begin{array}{cccccc} & & & & & 1 \\ & & & & & 1 & 1 \\ & & & & 1 & 2 & 1 \\ & & & 1 & 3 & 3 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{array}$$

Vous pouvez utiliser la règle de Pascal :

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Ceux qui veulent s'amuser : affichez le tableau aligné sous la forme donnée dans l'exemple.

Conversion (c'est le TP d'ASR1 ou quoi?! ☺)

Écrire une fonction de conversion d'un entier positif écrit en base décimale dans un nombre en base 2 (représenté dans un tableau). La fonction doit prendre en paramètre un entier et retourner un tableau. Exemple : pour le nombre 11 il faut obtenir

1	0	1	1
---	---	---	---

Factorielle (encore?)

On n'est plus dans les années '50 mais on n'est pas non plus chez les extraterrestres qui ont une puissance de calcul phénoménale. On souhaite calculer le factoriel d'un très grand nombre n . Le problème est que lorsque n est grand (1000 ou plus par exemple) son factoriel dépasse la taille mémoire allouée aux variables entières ou réelles... Par exemple :

$$100! = 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000.$$

Une idée c'est de décomposer le nombre n dans un tableau qui contient ses chiffres et de faire le traitement sur ce tableau plutôt que sur le nombre n .

1. Écrire une fonction qui prend en entrée un entier n et retourne un tableau de ses chiffres. Par exemple, pour le nombre 16528 la fonction devra retourner

1	6	5	2	8
---	---	---	---	---

2. En utilisant cette décomposition, écrire un programme de calcul de $n!$ pour tout n .

Fusion (et peut-être de la récursivité...)

1. Écrire un sous-programme qui réalise la fusion de deux tableaux triés en un tableau trié. Chaque élément des deux premiers tableaux doit se trouver dans le tableau résultat. La taille du tableau résultat est égale à la somme des tailles des deux premiers tableaux.
 - (a) Écrire une fonction qui trie deux tableaux d'entiers passés en paramètre, chacun de taille aléatoire comprise entre 10 et 15, avec des valeurs aléatoires comprises entre 1 et 20.
 - (b) Écrire une autre fonction qui fusionne les deux tableaux.Affichez les trois tableaux à la fin de l'exécution des deux fonctions.
2. QUESTION BONUS : En utilisant la fonction de fusion, implémentez l'algorithme de tri récursif par fusion.

Les tableaux multidimensionnels

Écrire une fonction qui calcule le produit de deux matrices d'entiers. L'utilisateur doit introduire les dimensions des matrices et leur contenu. Ensuite ces matrices seront passées en paramètres. La fonction devrait retourner une matrice résultat. Le produit matriciel sera également affiché ligne par ligne.