

TP 6: Boucles et tableaux (suite)

Informatique Fondamentale (IF121)

24 Novembre 2004

1 Suites numériques : encore et toujours

Exercice 1 : *Somme des puissances $k^{\text{èmes}}$*

Écrire une méthode qui, étant donnés deux entiers k et n , calcule la somme des puissances $k^{\text{èmes}}$ des n premiers entiers. Par exemple, `sommePuissances(3, 5) = 13 + 23 + 33 + 43 + 53 = 225`. (On réutilisera la méthode `puissance` du TP précédent.)

Exercice 2 : *Une suite*

Étant donné deux réels positif a et b , on définit la suite réelle $(x_n)_{n \in \mathbb{N}}$ de la manière suivante :

$$\begin{aligned}x_0 &= b \\ x_{i+1} &= \frac{x_i^2 + a}{2x_i}\end{aligned}$$

On admet que cette suite converge vers \sqrt{a} .

Écrire un programme qui lit un réel positif a et un entier n et qui renvoie une valeur approchée c de \sqrt{a} en utilisant l'approximation x_n . Afficher aussi `c*c` pour vérifier.

Exercice 3 : *Deux suites*

On définit une suite double :

$$\begin{aligned}u_0 &= 1 & u_{n+1} &= (u_n + v_n)/2 \\ v_0 &= 2 & v_{n+1} &= \sqrt{u_{n+1}v_n}\end{aligned}$$

On admet que les suites (u_n) et (v_n) sont adjacentes de limite $\sqrt{27}/\pi$. Écrire un programme qui lit un entier n et affiche l'approximation du nombre π obtenue à partir de v_n .

Note : la méthode `Math.sqrt` permet de calculer la racine carrée d'un nombre.

2 Traitement de données

Exercice 4 : *Recherche de « en »*

Écrire un programme qui lit une ligne de texte et qui compte le nombre de fois que cette ligne contient la lettre « e » suivie immédiatement de la lettre « n ». Par exemple, la phrase précédente contient 2 fois le fragment de texte « en », et celle-ci 4 fois.

Exercice 5 : *Palindrome*

Écrire un programme qui lit un mot et qui teste si ce mot est un palindrome, c'est-à-dire si on obtient le même mot en le lisant à l'envers. Par exemple, « kayak », « non », « ressasser », « selles » sont des palindromes ; « ânonna », « ressasse », « salles » n'en sont pas.

3 Tableaux de nombres

Exercice 6 : Norme, produit scalaire

Il est naturel de représenter le vecteur de coordonnées (x_0, \dots, x_{n-1}) (dans un espace vectoriel de dimension n) par le tableau dont ces n nombres sont les éléments.

(a) Écrire une méthode qui calcule le produit scalaire de deux vecteurs, dont on rappelle l'expression :

$$\vec{x} \bullet \vec{y} = \sum_{i=0}^{n-1} x_i y_i$$

(b) Écrire une méthode `norme` qui calcule la norme d'un vecteur. On rappelle que la norme du vecteur \vec{x} de coordonnées (x_0, \dots, x_{n-1}) est donnée par

$$\|\vec{x}\| = \sqrt{\vec{x} \bullet \vec{x}} = \sqrt{\sum_{i=0}^{n-1} x_i^2} = \sqrt{x_0^2 + x_1^2 + \dots + x_{n-1}^2}$$

Exercice 7 : Polynômes

Soit un polynôme à coefficients réels et de degré n :

$$P(X) = \sum_{i=0}^n a_i x^i = a_n X^n + a_{n-1} X^{n-1} + \dots + a_1 X + a_0$$

(a) Écrire une méthode `puissance` qui calcule la puissance $n^{\text{ème}}$ d'un nombre réel. (Cf. TP 5, ex. 1)

(b) À l'aide de cette méthode `evaluatePolynome`, écrire une méthode qui évalue un polynôme P en un point x : cette méthode prend en argument le tableau des coefficients de P et le nombre x , et renvoie le nombre $P(x)$.

(c) Si l'on suit le déroulement de l'exécution de la méthode `evaluatePolynome`, on observe qu'elle fait beaucoup de multiplications inutiles (quand on calcule x^{i+1} , on commence par recalculer x_i). Un moyen d'éviter cela est la *méthode de Horner*. Elle consiste à remarquer qu'on peut écrire

$$P(X) = (((\dots (a_n X + a_{n-1})X + \dots)X + a_2)X + a_1)X + a_0$$

Écrire une méthode `horner` qui évalue un polynôme en un point par cette technique.

4 Statistiques

Exercice 8 : Constitution d'un histogramme

Un examen est noté sur 10, par points entiers (c'est-à-dire que les notes possibles sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et 10). À fins statistiques, on veut savoir combien d'élèves ont chaque note. On suppose qu'il y a e étudiants, et qu'on dispose d'un tableau `notes` tel que `notes[i]` est la note du $i^{\text{ème}}$ étudiant. Construire un tableau `stats` tel que `stats[n]` contienne le nombre d'étudiants qui ont eu la note n .