

Info 1

Travaux dirigés - Série n°6

Les consignes données en tête de la série n° 2 restent bien sûr la règle pour ce qui suit :

Titre / Variables / Corps / Commentaires

Dans les algorithmes des applications utilisant des sous-programmes, écrire d'abord un algorithme général (programme principal) de l'application, ensuite des algorithmes détaillés (sous-programmes) des procédures et/ou fonctions.

Exercice 1. Puissance

- a) Écrire une fonction qui permet de calculer x^n , x étant un réel et n un entier.
- b) Écrire une procédure équivalente, qui soit affiche le résultat, soit le retourne dans un paramètre
- c) Écrire l'algorithme de l'application qui permet de répéter ce traitement en faisant appel à la fonction ou à la procédure précédentes avec des valeurs de x et n différentes, passées en paramètres, à la demande de l'utilisateur (en répondant à la question « voulez vous refaire le traitement ? O/N »).

Exercice 2. Puissance/Factorielle

Écrire l'algorithme général qui calcule $x^n/n!$

- a) En utilisant les fonctions puissance et factorielle
- b) En calculant directement (comment passe-t-on du terme en $(n-1)$ au terme en n ?)
- c) Écrire un algorithme qui répète le calcul de $x^n/n!$ En demandant les valeurs successives à l'utilisateur et en faisant appel aux fonctions

Note : Pour la méthode a), lors du passage sur machine, ne pas utiliser de grandes valeurs de n afin d'éviter des dépassements de capacité dus à la représentation des données.

Exercice 3. Représentation

- a. Écrire la fonction qui permet de calculer le nombre de chiffres nécessaire pour représenter un nombre donné en base 10 en un nombre dans une base b saisie en donnée (Exemple $b=2, 8, 16$).
- b. Écrire l'algorithme de l'application qui permet de répéter ce traitement à la demande de l'utilisateur. Utiliser dans ce cas une fonction qui retourne ce nombre de chiffres et qui utilise comme paramètres le nombre en base 10 et la base b .

Exercice 4. Fibonacci

- a. Écrire une procédure qui permet de calculer le $n^{\text{ème}}$ terme de la suite de Fibonacci, n étant passé en paramètre. Les termes de la suite de Fibonacci sont obtenus en calculant la somme des deux termes précédents : $T_0=1, T_1=1, T_n = T_{n-1} + T_{n-2}$ (pour $n \geq 2$).
- b. Écrire l'algorithme de l'application qui permet de calculer différentes suites de Fibonacci, ce calcul s'arrêtera à la demande de l'utilisateur. Utiliser une procédure qui permet d'afficher les n termes d'une suite de Fibonacci. Le nombre de termes à afficher est saisi en donnée et passé en paramètre à la procédure.