

TP 1 : récursivité

Conseil : pensez à organiser correctement vos fichiers (répertoires/ sous-répertoires correspondant aux différents TPs). Utilisez un *Makefile* pour la compilation, même si la plupart du temps un seul fichier doit être compilé : la méthodologie s'apprend au fur et à mesure !

Exercice 1

Écrire un programme qui calcule récursivement et affiche la factorielle d'un entier strictement positif, lu au clavier.

Modifier ce programme (en introduisant une variable locale à la fonction récursive) pour afficher la valeur du paramètre avant chaque appel récursif, et la valeur du résultat délivré par chaque appel récursif. Vérifier les affichages produits.

Exercice 2

Écrire un programme qui calcule récursivement et affiche le nombre de chiffres d'un entier strictement positif, lu au clavier. Exemple d'exécution :

Donnez un entier positif : 1789

Ce nombre possède 4 chiffres.

Exercice 3

Écrire un programme qui lit un entier n puis calcule récursivement et affiche la $n^{\text{ème}}$ valeur de la suite de Fibonacci. Faire afficher, au début de la fonction récursive, la valeur du paramètre, et vérifier que les appels se déroulent correctement.

Écrire les deux versions itératives de ce programme (avec et sans tableau). Comparer les affichages produits avec ceux de la version récursive.

Exercice 4

On appelle "palindrome" un mot ou une phrase qui se lit de la même façon à l'endroit comme à l'envers, sans tenir compte des espaces.

Exemple : le mot "ABCBA" est un palindrome.

La phrase "ESOPE RESTE ET SE REPOSE" (sans les espaces on obtient "ESOPERESTEETSEREPOSE") en est également un

Écrire un programme utilisant une fonction récursive qui détermine si la séquence de caractère saisie au clavier par l'utilisateur est un palindrome.

Exercice 5

Écrire un programme de **tri rapide** sur un tableau d'entiers ($N = 20$) initialisé par affectation :

- écrire les fonctions *permuter*, *partition* et *tri_rapide*,
- écrire enfin le programme principal.

Ajouter des affichages dans *tri_rapide* pour vérifier le bon déroulement du tri :

- afficher les indices de début et de fin du sous-tableau à trier et la valeur du pivot ($T[\text{debut}]$) juste avant l'appel de la fonction *partition*,
- afficher le contenu du tableau juste après l'appel de la fonction *partition*.

Pour observer le comportement de l'algorithme, il est préférable de remplir le tableau avec des valeurs rangées *de manière aléatoire*.

Question complémentaire : écrire la version itérative du tri rapide, à l'aide d'une pile.