

Exercice 1. Manipulation d'entiers

Parmi les nombres entiers compris entre 0 et 500, certains ont la propriété suivante: ils sont la somme des cubes de leurs chiffres en base 10. Soit un entier n compris entre 0 et 500, il s'écrit $n = xyz$ en base 10 (x est le nombre des centaines, y des dizaines et z celui des unités). Il a la propriété recherchée si et seulement si $n = x^3 + y^3 + z^3$.

Ecrire un programme qui affiche tous ces nombres (entre 0 et 500). On pourra utiliser une fonction prenant trois arguments passés par référence (i.e. par adresse), qui permet de retourner ainsi le chiffre des centaines, des dizaines et des unités d'un nombre entier n passé en argument.

Exercice 2. Manipulation de flottants

Soit la suite (U_n) définie pour n positif par $U_n = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$, (plus précisément par $U_0 = 1$, $U_2 = 1 + 1/2$ et $U_n = U_{n-1} + 1/n$). Cette suite porte le nom de série harmonique. Cette suite tend vers l'infini quand n tend vers l'infini. On devrait donc, si on imprime ses termes, les voir sans cesse augmenter. Mais sur un ordinateur, avec des flottants, la précision n'est pas suffisante pour distinguer les termes à partir d'un certain rang.

Écrire un programme C qui calcule pas à pas les termes U_n de cette suite, et affiche la valeur d'un terme toutes les mille itérations (ou plus si votre machine a 32 ou 64 de bits) en utilisant le type float pour stocker les termes. On peut obtenir pour les premières valeurs l'affichage suivant

1000 : 7.485478

2000 : 8.178369

...

Et la suite finit par devenir stationnaire. Interrompez l'affichage si le programme vous semble trop long. Augmentez le nombre d'itérations jusqu'à constater que la valeur calculée est stationnaire. Recommencez ensuite le procédé, mais en utilisant cette fois des double pour encoder les termes du calcul. Pour apprécier la différence de précision, diminuez la fréquence d'affichage, et utilisez des entiers plus longs pour augmenter le nombre d'itérations. Interrompez le calcul si vous voyez que les temps nécessaires sont trop long pour poursuivre l'expérience.

Exercice 3. Tri, impression et initialisation d'un tableau d'entiers.

Ecrire un programme qui déclare un tableau par avec `tableau = {9,8,3,-1,10,0}` par exemple, et utilise ensuite une macro pour calculer son nombre d'éléments. On écrira aussi une procédure générale d'impression de tableau, et une procédure de tri que l'on testera dans le main. Ecrire ensuite une procédure permettant d'initialiser un tableau dont la taille et les éléments seront entrés par l'utilisateur. Tester votre procédure d'initialisation et à nouveau celles d'impression et de tri.

Rappel: la fonction `void* malloc(sizeof(int)*nb)` permet d'allouer dynamiquement la place mémoire nécessaire pour l'allocation d'un tableau d'entiers de nb éléments.