Exercices à faire pendant le TP 8

Recherche des diviseurs d'un entier

Un nombre entier positif est un nombre premier s'il ne possède comme seuls diviseurs que le nombre 1 et lui-même. Par ailleurs, un nombre entier positif est dit parfait s'il est égal à la somme de ses diviseurs, lui-même étant exclu. Par exemple 28 est parfait car $28 = 1 + 2 + 4 + 7 + 14$.
Notre but est d'écrire un programme qui :
- lit un nombre n au clavier, avec $n < M$, où M est une constante du programme ;
- recherche et affiche les diviseurs de n ;
- indique si n est premier ;
- indique si n est parfait ;
- recherche tous les nombres parfaits inférieurs à un entier t ($< M$) entré au clavier par l'utilisateur.

Le programme utilise une variable globale D, qui est un tableau de M entiers, et dont les cases contiendront les diviseurs successifs de l'entier n.
1) Ecrivez la fonction `int diviseur(int a, int b)` qui renvoie 1 si a est un diviseur de b, et 0 sinon.
2) Ecrivez la fonction `int tous_les_diviseurs(int p)` qui remplit le tableau D avec les diviseurs de p hors 1 et p, et qui retourne le nombre de ces diviseurs (cette fonction appelle la fonction *diviseur*).
3) Ecrivez une fonction `int premier (int a)` qui renvoie 1 si a est premier, et 0 sinon.
4) Ecrivez une fonction `int parfait(int a)` qui renvoie 1 si a est parfait, et 0 sinon.
5) Ecrivez le programme principal qui saisit un nombre entier n donné par l'utilisateur, en s'assurant que ce nombre est compris dans l'intervalle [1,M[, appelle les fonctions écrites ci-dessus et répond au problème posé.

Suite de Fibonacci

Ecrivez un programme qui permet de calculer le terme d'ordre n de la suite de Fibonacci définie par la relation de récurrence suivante :
$U_0 = 0 \qquad U_1 = 1 \qquad$ et, pour tout $n \geq 2$, $U_n = U_{n-1} + U_{n-2}$

Calcul d'une racine carrée avec l'algorithme de Héron

L'algorithme de Héron permet de calculer une valeur approchée de $\sqrt{a}$, pour $a > 0$, de la façon suivante : $\sqrt{a}$ est la limite de la suite u_n définie par

$$u_{n+1} = \frac{1}{2}(u_n + \frac{a}{u_n})$$

où l'on donne à u_0 une valeur positive quelconque.
En utilisant cette méthode, écrivez un programme qui, pour une valeur $a > 0$ donnée par l'utilisateur, calcule une valeur v approchée de $\sqrt{a}$ telle que $v^2 - a \leq 10^{-8}$.
Vous pourrez par exemple prendre $u_0 = 1$.
Vous comparerez v avec la valeur donnée par la fonction $sqrt(x)$ du C (bibliothèque math.h).

S'il vous reste du temps en TP : **programmez l'addition binaire**

Reprenez ce que vous avez fait au TP 7 pour calculer la représentation binaire d'un entier. Ou faites cet exercice du TP7 si vous n'en avez pas eu le temps précédemment.
Puis écrivez un programme qui :
- saisit deux nombres entiers positifs a et b inférieurs à 100 ;
- calcule les représentations binaires de a et b et les enregistre dans les tableaux Ta et Tb ;
- effectue l'addition binaire de ces nombres, en enregistrant le résultat dans un tableau Ts ;
- calcule la valeur décimale v du nombre binaire contenu dans Ts ;
- vérifie que v est bien égal à $a + b$.