

TRAVAUX PRATIQUES MAPLE NO. 2 (DEUXIÈME TRIMESTRE)

CPES FEYDER II

EXERCICES D'ARITHMÉTIQUE (SUITE)

Exercice 1 :

- (1) Écrire une procédure `somme` qui prend comme argument un entier naturel n et qui calcule la somme des chiffres composant n . Par ex., `somme(28) = 2 + 8 = 10` car le nombre 28 est composé des chiffres « 2 » et « 8 » ;
- (2) Tester la procédure sur les entiers $n = 0$, puis 1, 9, 123456789 et 100! ;
- (3) Écrire une procédure `itereSomme` qui prend comme argument un entier naturel n et qui itère la procédure précédente, tant que le nombre obtenu comprend plusieurs chiffres. Par ex., pour $n = 28$, on calcule tout d'abord `somme(28) = 10` puis on recommence (puisque 10 a deux chiffres), `somme(10) = 1 + 0 = 1`. On arrête étant donné que le résultat obtenu s'écrit sur un seul chiffre ;
- (4) Tester cette procédure avec les mêmes entiers que la question 2.

Exercice 2 :

- (1) Écrire une procédure `facteurs` qui prend comme argument un entier naturel non nul n et qui renvoie la liste de ses facteurs premiers : par ex., `facteurs(50)` va renvoyer la liste `[2, 5, 5]` puisque $50 = 2 \times 5^2$. Pour coder cette procédure vous n'utiliserez pas la commande `ifactor` ;
- (2) Tester la procédure sur les exemples suivants :
 - `facteurs(360)` doit renvoyer `[2, 2, 2, 3, 3, 5]` ;
 - `facteurs(81)` doit renvoyer `[3, 3, 3, 3]` ;
 - `facteurs(13)` doit renvoyer `[13]` ;
 - `facteurs(1)` doit renvoyer `[]` ;

Exerice 3 : On se propose de coder l'algorithme d'Euclide qui permet de déterminer le PGCD d des entiers a et b .

L'algorithme d'Euclide est défini par la suite des divisions euclidiennes :

$$\begin{aligned}
 a &= bq_1 + r_1 \\
 b &= r_1q_2 + r_2 \\
 r_1 &= r_2q_3 + r_3 \\
 r_2 &= r_3q_4 + r_4 \\
 &\dots \\
 r_n &= r_{n+1}q_{n+2} + r_{n+2} \\
 r_{n+1} &= r_{n+2}q_{n+3} + 0 .
 \end{aligned}$$

Alors $d = \text{pgcd}(a, b) = r_{n+2}$.

- (1) Écrire une procédure `pgcd` (sans utiliser la commande `igcd`) qui prend comme arguments deux entiers a et b et qui renvoie leur PGCD calculé selon l'algorithme d'Euclide ;
- (2) Tester la procédure en calculant `pgcd(16, 15)`, `pgcd(48, 4)`, `pgcd(4, 48)`, `pgcd(123456789, 987654321)` et `pgcd(123456789987654321, 159753258456951753852654)` et en vérifiant chaque résultat obtenu à l'aide de la commande `ifactor`.