

TP : représentation des entiers

Informatique tronc commun

1 Changement de base

On rappelle que pour convertir un entier $n_{p-1}...n_1n_0b$ d'une base b en base 10 il suffit de calculer $n_0 + bn_1 + b^2n_2 + \dots + b^{p-1}n_{p-1}$. Par exemple $121_3 = 1 \times 3^2 + 2 \times 3 + 1 = 16 (= 16_{10})$.

1. Écrire une fonction `to_base10` ayant comme arguments une base `b` et une liste `L`, et renvoyant le nombre $L[\text{len}(L) - 1]...L[1]L[0]_b$ converti en base 10. Par exemple, `to_base10(3, [1, 2, 1])` doit renvoyer 121_3 en base 10, c'est à dire 16.

On rappelle que pour connaître les chiffres d'un entier n en base b on peut écrire sa division euclidienne par $b : n = bq + r$. Le reste r est alors le premier chiffre de n en base b . Il suffit alors d'appliquer à nouveau la méthode en remplaçant n par q , tant que $n \neq 0$.

Par ex., pour convertir 26 en base 2, on effectue des divisions par 2 jusqu'à obtenir 0 :

$$\begin{array}{r}
 26 \mid 2 \\
 \underline{0} \mid 13 \mid 2 \\
 \quad \underline{1} \mid 6 \mid 2 \\
 \quad \quad \underline{0} \mid 3 \mid 2 \\
 \quad \quad \quad \underline{1} \mid 1 \mid 2 \\
 \quad \quad \quad \quad \underline{1} \mid 0
 \end{array}$$

On en déduit alors que $26 = 11010_2$.

2. Écrire une fonction `from_base10` ayant une base `b` et un nombre `n` en arguments et renvoyant les chiffres de `n` en base `b`.

2 Opérations

On rappelle que l'on peut additionner deux nombres écrits en base b en ajoutant les chiffres un par un, en gardant une retenue de 1 si la somme est supérieure ou égale à b . Voici deux exemples, en base 10 puis en base 2 :

$$\begin{array}{r}
 \overset{1}{3} \overset{1}{2} \overset{1}{8} \\
 + 974 \\
 \hline
 1302
 \end{array}
 \qquad
 \begin{array}{r}
 \overset{1}{1} \overset{1}{0} \overset{1}{0} \overset{1}{1} \subscript{2} \\
 + \quad 11 \subscript{2} \\
 \hline
 1100 \subscript{2}
 \end{array}$$

On représente un nombre $n = n_{p-1}...n_1n_0b$ en base b par la liste `L` de ses chiffres, où `L[i] = ni` est le i ème chiffre de n . Par exemple 1010_2 est représenté par `[0, 1, 0, 1]`.

1. Écrire une fonction `add(L1, L2, b)` renvoyant la liste des chiffres de la somme de `L1` et `L2`, en base b . Par exemple `add([1, 0, 0, 1], [1, 1], 2)` doit renvoyer `[0, 0, 1, 1]`. Vérifier votre fonction, éventuellement en utilisant `to_base10` et `from_base10`.

On rappelle que l'on peut multiplier deux nombres écrits en base b de la façon suivante :

$$\begin{array}{r}
 1001 \subscript{2} \\
 \times \quad 11 \subscript{2} \\
 \hline
 1001 \subscript{2} \\
 + 10010 \subscript{2} \\
 \hline
 11011 \subscript{2}
 \end{array}$$

2. Écrire une fonction `mult(L1, L2, b)` renvoyant la liste des chiffres de la multiplication de `L1` et `L2`, en base b . Par exemple `mult([1, 0, 0, 1], [1, 1], 2)` doit renvoyer `[1, 1, 0, 1, 1]`. On pourra réutiliser `add`.
3. Écrire aussi des fonctions pour soustraire/diviser deux nombres écrits en base b .