

Système de Numération et Changement de Bases

Généralités

D'une manière générale, le nombre $a_n a_{n-1} \dots a_2 a_1 a_0$, s'il est exprimé en base B, ne pourra comporter que des chiffres a_i compris entre 0 et B - 1. Ainsi, en base 10 nous disposons des 10 chiffres arabes de 0 à 9, en base 2 nous ne pourrions utiliser que les chiffres 0 et 1 ; en base 16 : 0, 1, 2, ..., 9 et nous continuerons avec les lettres A, B, C, D, E, F qui auront les valeurs dix, onze, douze, treize, quatorze, quinze dans notre base usuelle.

Au nombre $a_n a_{n-1} \dots a_2 a_1 a_0$ sera associé la valeur $a_n * B^n + a_{n-1} * B^{n-1} + \dots + a_2 * B^2 + a_1 * B + a_0$ calculée en base 10.

Expression en base 10 d'un nombre exprimé en base B

Comme nous sommes habitués à calculer en base 10, nous pouvons effectuer tout changement de bases en effectuant un passage par la base 10. La seule exception à cette règle est pour les passages entre les bases 2 et 16.

Les notations se feront à partir de la règle :

$$(a_n a_{n-1} \dots a_2 a_1 a_0)_B = (a_n * B^n + a_{n-1} * B^{n-1} + \dots + a_2 * B^2 + a_1 * B + a_0)_{10}$$

Les calculs pourront être exprimés à l'aide de la méthode de Hörner (gage de vitesse et de précision) :

$$a_n * B^n + a_{n-1} * B^{n-1} + \dots + a_2 * B^2 + a_1 * B + a_0 = (((a_n * B + a_{n-1}) * B + \dots) + a_2) * B + a_1 * B + a_0$$

Expression en base B d'un nombre X exprimé en base 10

D'un nombre $X = (((a_n * B + a_{n-1}) * B + \dots) + a_2) * B + a_1 * B + a_0$, nous déduisons simplement que lorsqu'on divise (division cartésienne ou entière) X par B, on obtient un reste égal à a_0 et un quotient entier égal à

$$Y = (((a_n * B + a_{n-1}) * B + \dots) + a_2) * B + a_1.$$

De même, en effectuant la division entière de Y par B, nous obtenons un reste égal à a_1 et un quotient égal à $Z = ((a_n * B + a_{n-1}) * B + \dots) + a_2$. En opérant les divisions successives, nous obtenons tous les chiffres du nombre (on arrête les divisions lorsque le quotient obtenu est nul).

Passage entre les bases 2 et 16

En base 16	En base 2	En base 10
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Ce tableau montre tout l'intérêt de la base 16. Comme exprimer les nombres en base 2 est particulièrement fastidieux et sujet à erreur, l'idée a été de regrouper par 4 et de convertir chaque groupe en 1 chiffre hexadécimal. Les chiffres de la base 16 permettent en effet de trouver le chiffre correspondant à chaque groupe de 4 chiffres binaires.

Le passage de la base 2 à la base 16 se fera donc par regroupement de 4 chiffres binaires consécutifs (en partant de la droite et en allant vers la gauche) et de la traduction en base 16 à l'aide du tableau décrit ci-contre.

$$\text{Exemple : } (111011001)_2 = 0001\ 1101\ 1001 = (1D9)_{16}$$

Le passage de la base 16 à la base 2 se fera par traduction d'un chiffre hexadécimal en 4 chiffres binaires à l'aide du tableau ci-contre.

$$\text{Exemple : } (3AF)_{16} = 0011\ 1010\ 1111 = (1110101111)_2$$

Exercice :

Vous pourrez utiliser dans les fonctions à écrire la chaîne de caractères donnée en tout début du programme principal : lettres = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'. Cette chaîne a comme particularité que tous ses caractères peuvent être utilisés dans l'écriture d'un nombre en une base inférieure à 36 mais peuvent aussi être obtenus suivant leur indice de position dans la chaîne lettres. Ainsi, le nombre $(1A)_{16}$ donné comme la chaîne de caractères st='1A' a comme valeur en base 10 : $(\text{strindex}(\text{lettres}, \text{part}(\text{st}, 1)) - 1) * 16 + (\text{strindex}(\text{lettres}, \text{part}(\text{st}, 2)) - 1)$.

1) Ecrire une fonction de_base_en_b10 qui reçoit en paramètres une chaîne de caractères qui est l'expression d'un nombre en une base quelconque et comme autre paramètre la base depuis laquelle le nombre doit être transformé et renvoie l'entier résultat.

2) Ecrire une fonction de_b10_en_base qui reçoit en paramètres un nombre en base 10 et comme autre paramètre la base dans laquelle ce nombre doit être transformé. Le résultat sera la chaîne de caractère donnant l'expression du nombre dans la base demandée.

3) Ecrire une fonction de_base1_en_base2 transformant un nombre d'une base 1 en une base 2.

```
function [n]=de_base_en_b10(st,base)
n=0;
for i=1:length(st) do
b=strindex(lettres,part(st,i))
n=n*base+(b(1)-1)
end
```

```
function [st]=de_b10_en_base(n,base)
st="";
while n<>0 do
i=modulo(n,base);
st=part(lettres,i+1)+st;
n=(n-i)/base;
end
```

```
function [st2]=de_base1_en_base2(st1,b1,b2)
st2=de_b10_en_base(de_base_en_b10(st1,b1),b2)
```