

MÉTHODES DE CONVERSION :

BASE 2/10

1°) Passage de la base 2 à la base 10 : c'est la définition !!!

Un exemple pour rappel $(10011101)_2 =$

On peut vite dépasser les capacités calculatoires « classiques »

2°) Passage de la base 10 à la base 2

**METHODE : on divise par 2 le nombre donné , on note le quotient et le reste (qui est 0 ou 1) puis on divise par 2 le quotient et on note le reste (0 ou 1)....on s'arrête quand le quotient est 0.
On déroule ensuite les restes**

EXEMPLE a suivre avec le professeur : $(435)_{10}$

Ainsi $(435)_{10} = (110110011)_2$

NB : on remarque qu'ici , il faut AU MINIMUM 9 bits pour le codage. Ainsi, cet entier codé sur 4 ou 8 bits donne le message « *overflow* » qui informe que la capacité d'écriture est insuffisante

ALLER PLUS LOIN : seriez vous capable de programmer cela en Python ?

BASE 10/16

1°) Passage de la base 16 à la base 10 : c'est la définition !!!

Un exemple pour rappel $(40 A 7 D)_{16} =$

On peut vite dépasser les capacités calculatoires « classiques »

2°) Passage de la base 10 à la base 16

**METHODE : on divise par 16 le nombre donné , on note le quotient et le reste (qui est ENTRE 0 et 15) puis on divise par 16 le quotient et on note le reste (0 à 15)...on s'arrête quand le quotient est 0.
On déroule ensuite les restes**

EXEMPLE a suivre avec le professeur : $(435)_{10}$

Ainsi $(435)_{10} = (1B3)_{16}$

ALLER PLUS LOIN : seriez vous capable de programmer cela en Python ?

BASE 2/16

1°) On peut chaque fois repasser par la base 10 (de référence)...en utilisant les méthodes évoquées ci dessus. C'est LONG et source d'erreur !! Essayez vous verrez !

2°) Passage de la base 2 à la base 16 : il suffit de grouper les chiffres binaires par 4 (c'est pourquoi la base 16 est souvent utilisée pour simplifier l'écriture des nombres binaires)

$$(1001010111110011)_2 = (95F3)_{16} \text{ car } \begin{array}{cccc} 1001 & 0101 & 1111 & 0011 \\ 9 & 5 & F & 3 \end{array} \quad 5 \quad (\text{on peut utiliser la table ci dessous})$$

3°) Passage de la base 16 à la base 2

En complétant le tableau de correspondance suivant,

Nombre en hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Nombre en binaire	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Le travail inverse est donc aisé :

$$(E5BB9)_{16} = (11100101101110111001)_2 \text{ puisque } \begin{array}{cccc} E & 5 & B & B & 9 \\ 1110 & 0101 & 1011 & 1011 & 1001 \end{array}$$

ALLER PLUS LOIN : seriez vous capable de programmer cela en Python ?

CONVERSIONS DE BASE EN PYTHON

1°) Elaborer une fonction Python qui réalise une conversion d'un nombre n dans une base b.

2°) Les concepteurs de Python sont géniaux....Des fonctions existent en Python qui réalisent toutes ces conversions : **la base par défaut est la base 10, les entiers illimités dans les bases 2,10 ou 16 (pas d'overflow)**

Entrez dans la console de Python la série d'instructions suivantes en validant par . Observez les résultats obtenus puis complétez le tableau « aide mémoire ».

Manipuler une séquence de nombres dans différentes bases :

<i>Pour ...</i>	<i>Je tape dans la console</i>
Pour convertir un nombre binaire en base 10	
Obtenir l'écriture décimale du nombre binaire 01001101	>>> OU >>>
Pour convertir un nombre décimal en base 2	
Obtenir l'écriture binaire de 43 (écrit en décimal)	>>>
Pour convertir un nombre hexadécimal en base 10	
Obtenir l'écriture décimale de DF40E (écrit en hexadécimal)	>>> OU >>>
Pour convertir un nombre décimal en base 16	
Obtenir l'écriture hexadécimale de 1759 (écrit en décimal)	>>>
Pour convertir un nombre binaire en base 16	
Obtenir l'écriture hexadécimale de 1001011(écrit en binaire)	>>>
Pour convertir un nombre hexadécimal en base 2	
Obtenir l'écriture binaire de A48E (écrit en hexadécimal)	>>> OU >>>