

# Thème: Binaire - Circuit - Codage

## #2 Codage des nombres entiers

### Sommaire

<b>Sommaire</b>	<b>1</b>
<b>Les systèmes de numération en base n</b>	<b>2</b>
<b>Mission 2.1.</b>	<b>2</b>
<b>Convertisseur base n - décimal</b>	<b>3</b>
<b>Mission 2.2.</b>	<b>3</b>
<b>Mission 2.3.</b>	<b>3</b>
<b>Convertisseur décimal - binaire</b>	<b>4</b>
<b>Mission 2.4.</b>	<b>4</b>
<b>Mission 2.5.</b>	<b>5</b>

L'objectif de cette activité est de:

- comprendre comment manipuler la numération en base n
- programmer la conversion base n - décimal
- programmer la conversion décimal - binaire

# Les systèmes de numération en base n

Rappelez vous comment sont construits les nombres dans le système décimal (base 10) qui dispose de 10 chiffres ( 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9):

$$\begin{aligned} 1237 &= 1000 + 200 + 30 + 7 \\ &= 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 7 \times 10^0 \end{aligned}$$

Mais on peut aussi exprimer un nombre dans un système base  $n$  (avec  $n$  un entier naturel) qui dispose de  $n$  chiffres

Les systèmes de numération les plus utilisées sont:

- le système binaire (base 2) qui dispose de 2 chiffres: 0 et 1
- le système hexadécimal (base 16) qui dispose de 16 chiffres/caractères: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F

Ainsi 1101 en base 2 est égal à 13 en base 10:

$$\begin{aligned} (1101)_2 &= (1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)_{10} \\ &= (8 + 4 + 0 + 1)_{10} \\ &= (13)_{10} \end{aligned}$$

Et 1A0F en base 16 est égal à 6671 en base 10:

$$\begin{aligned} (1A0F)_{16} &= (1 \times 16^3 + 10 \times 16^2 + 0 \times 16^1 + 15 \times 16^0)_{10} \\ &= (4096 + 2560 + 0 + 15)_{10} \\ &= (6671)_{10} \end{aligned}$$

#####

## Mission 2.1.

- Calculez la valeur décimale de  $(10101010)_2$  et  $(3E1A)_{16}$
- Combien de nombre peut-on coder sur un octet ? Quel est le plus grand?  
NB: un octet = 8 bits , 1 bit (**binary digit**) étant une information binaire
- Combien faut-il de 'chiffres' en hexadécimal pour coder l'équivalent de 4 octets en binaire?

#####

# Convertisseur base n - décimal

#####

## Mission 2.2.

Donnez un algorithme en langage naturel qui convertit un nombre binaire en décimal.

#####

#####

## Mission 2.3.

Programmez une fonction en Python qui convertit un nombre en base n en décimal:

- Entrée:
  - n: entier représentant la base de départ ( n'importe quel entier  $2 \leq n \leq 10$  )
  - le nombre en base n
- Sortie:
  - le nombre équivalent en base 10

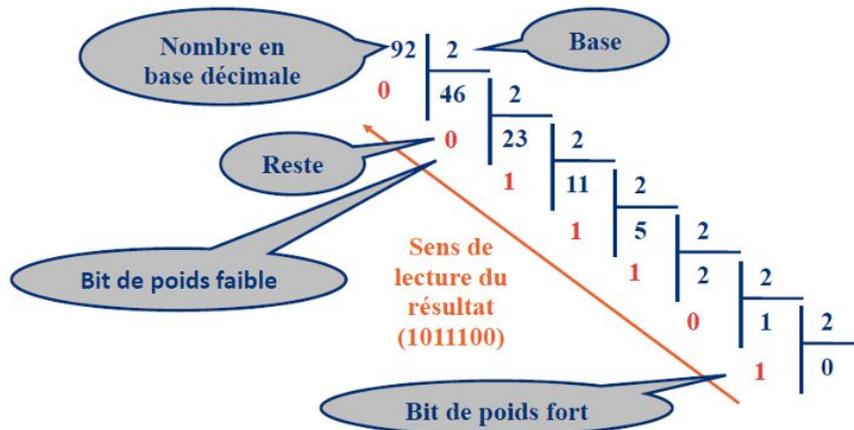
En ++ (optionnel): généraliser à une base  $> 10$

#####

# Convertisseur décimal - binaire

Pour convertir un nombre en base n vers le binaire, on peut faire des divisions euclidiennes successives.

Par exemple, en base 2, on procède comme suit:



$$\begin{aligned}
 (92)_{10} &= (2 \times 46 + 0)_{10} \\
 &= (2 \times (2 \times 23 + 0) + 0)_{10} \\
 &= (2 \times (2 \times (2 \times 11 + 1) + 0) + 0)_{10} \\
 &= (2 \times (2 \times (2 \times (2 \times 5 + 1) + 1) + 0) + 0)_{10} \\
 &= (2 \times (2 \times (2 \times (2 \times (2 \times 2 + 1) + 1) + 1) + 0) + 0)_{10} \\
 &= (2 \times (2 \times (2 \times (2 \times (2 \times (2 \times 1 + 0) + 1) + 1) + 1) + 0) + 0)_{10} \\
 &= (2 \times (2 \times (2 \times (2 \times (2 \times (2 \times 0 + 1) + 0) + 1) + 1) + 1) + 0) + 0)_{10} \\
 &= (1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0)_{10} \\
 &= (1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0)_{2}
 \end{aligned}$$

#####

## Mission 2.4.

Donnez un algorithme en langage naturel qui convertit un nombre décimal en binaire.

#####

#####

## Mission 2.5.

Programmez une fonction en Python qui convertit un nombre décimal en binaire:

- Entrée:
  - le nombre décimal
- Sortie:
  - un texte avec des 0 et 1 représentant le nombre binaire

En ++ (optionnel):

- généraliser à une base  $n$  quelconque
- programmez un convertisseur base  $n$  à base  $m$  pour tout entier  $n$  et  $m$

#####