Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 → Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- □ Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  ⇒ le chiffre le plus à gauche est 1

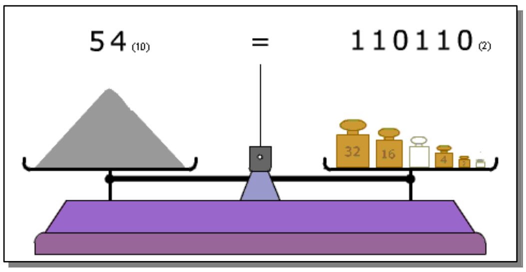
  Reste à représenter 420 – 1x16^2 = 164 unités

- □ On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 ⇒ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- □ Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.



Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.

Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = B^0 = 1 = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : $\mathbf{111\ 0010\ 0110_{2}}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	1×16^2 = 7 x 256
0		C'est fini, il ne reste plus rien à diviser		

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

EXERCICES

Rechercher par la méthode de Hornner :

203_8 = ...　　　　101010_2 = ...　　　　$20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$　　　　COCA$_{16}$ = ... $_{10}$

100_{10} = ... $_{2}$　　　　1011011_2 = ... $_{10}$

100_{10} = ... $_{16}$　　　　236_8 = ... $_{10}$

1023_{10} = ... $_{16}$　　　　FFF$_{16}$ = ... $_{10}$

1023_{10} = ... $_{2}$

Conversion d'un nombre décimal entier vers une base B quelconque

Rappel : valeur de chaque chiffre

Base 10 $\rightarrow$ Base B

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- □ Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

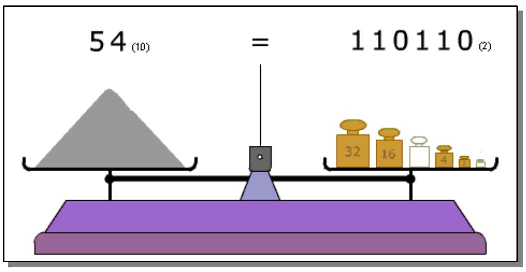
  Reste à représenter 420 – 1x16^2 = 164 unités

- □ On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- □ Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.



Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.

Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = $B^0 = 1$ = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire $\Rightarrow$ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : $\mathbf{111\ 0010\ 0110_{2}}$

Convertir **1830** en hexadécimal $\Rightarrow$ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	$1 \times 16^2 = 7 \times 256$
0		C'est fini, il ne reste plus rien à diviser		

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10
« Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

$$\begin{array}{cccc} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ & 2 & 6 & 12 \\ \hline & 3 & 6 & 13 \end{array}$$

123_8 = (((**1** x 8) + **2**) x 8) + **3**

$$\begin{array}{ccc} 1 & 2 & 3 \\ & 8 & 80 \\ \hline & 10 & 83 \end{array}$$

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

$$\begin{array}{ccc} \mathbf{2} & \mathbf{0} & \mathbf{12} \\ & 32 & 64 \\ \hline & 32 & 76 \end{array}$$

Exercices

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$	$COCA_{16}$ = ... $_{10}$
100_{10} = ... $_{2}$	1011011_2 = ... $_{10}$
100_{10} = ... $_{16}$	236_8 = ... $_{10}$
1023_{10} = ... $_{16}$	FFF_{16} = ... $_{10}$
1023_{10} = ... $_{2}$	

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

  Reste à représenter 420 – 1x16^2 = 164 unités

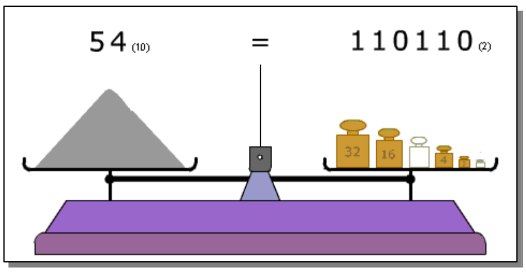
- On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l'exercice « [Pesée d'un nombre binaire]» pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = B^0 = 1 = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : $\mathbf{111\ 0010\ 0110_2}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	1×16^2 = 7 x 256
0		C'est fini, il ne reste plus rien à diviser		

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
    Multiplier par la base
    Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

```
1    1    0    1
     2    6    12
     3    6    13
```

123_8 = (((**1** x 8) + **2**) x 8) + **3**

```
1    2    3
     8    80
     10   83
```

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

```
2    0    12
     32   64
     32   76
```

EXERCICES

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$	$COCA_{16}$ = ... $_{10}$
100_{10} = ... $_{2}$	1011011_2 = ... $_{10}$
100_{10} = ... $_{16}$	236_8 = ... $_{10}$
1023_{10} = ... $_{16}$	FFF_{16} = ... $_{10}$
1023_{10} = ... $_{2}$	

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)

- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)

- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)

- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire

- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

  Reste à représenter 420 – $1\text{x}16^2$ = 164 unités

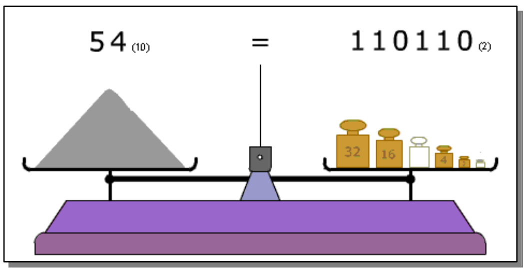
- On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = B^0 = 1 = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir **1830_{10}** en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : **$111\ 0010\ 0110_2$**

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	1×16^2 = 7 x 256
0		C'est fini, il ne reste plus rien à diviser		

Résultat : **726_{16}** (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d’une base B en base 10
« Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d’un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l’algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu’il reste des chiffres à droite
{
    Multiplier par la base
    Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

EXERCICES

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$ $COCA_{16}$ = ... $_{10}$
100_{10} = ... $_{2}$ 1011011_2 = ... $_{10}$
100_{10} = ... $_{16}$ 236_8 = ... $_{10}$
1023_{10} = ... $_{16}$ FFF_{16} = ... $_{10}$
1023_{10} = ... $_{2}$

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

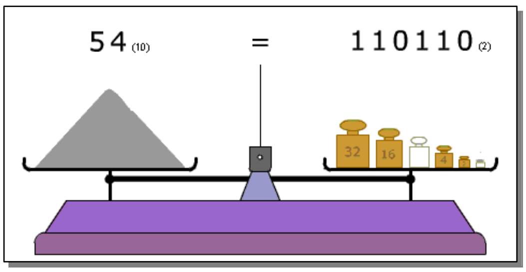
  Reste à représenter $420 - 1\text{x}16^2 = 164$ unités

- On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.



Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.

Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = $B^0 = 1$ = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : $\mathbf{111\ 0010\ 0110_2}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	$1 \times 16^2 = 7 \times 256$
0		C'est fini, il ne reste plus rien à diviser		

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d’une base B en base 10
« Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d’un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l’algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu’il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

<u>EXERCICES</u>

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$

100_{10} = ... $_{2}$

100_{10} = ... $_{16}$

1023_{10} = ... $_{16}$

1023_{10} = ... $_{2}$

$COCA_{16}$ = ... $_{10}$

1011011_2 = ... $_{10}$

236_8 = ... $_{10}$

FFF_{16} = ... $_{10}$

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- □ Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

  Reste à représenter 420 – 1x16^2 = 164 unités

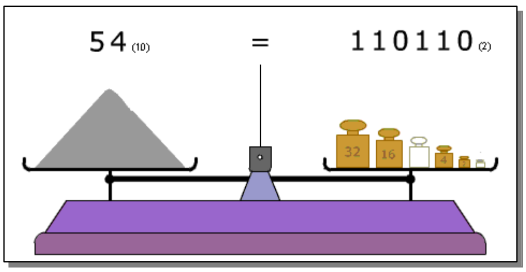
- □ On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- □ Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = B^0 = 1 = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir **1830_{10}** en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : **$111\ 0010\ 0110_2$**

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	1×16^2 = 7 x 256
0		C'est fini, il ne reste plus rien à diviser		

Résultat : **726_{16}** (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d’une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d’un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l’algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu’il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

<u>EXERCICES</u>

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$
100_{10} = ... $_{2}$
100_{10} = ... $_{16}$
1023_{10} = ... $_{16}$
1023_{10} = ... $_{2}$

$COCA_{16}$ = ... $_{10}$
1011011_2 = ... $_{10}$
236_8 = ... $_{10}$
FFF_{16} = ... $_{10}$

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 → Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)

- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)

- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)

- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire

- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  ⇒ le chiffre le plus à gauche est 1

  Reste à représenter 420 – $1 \text{x} 16^2$ = 164 unités

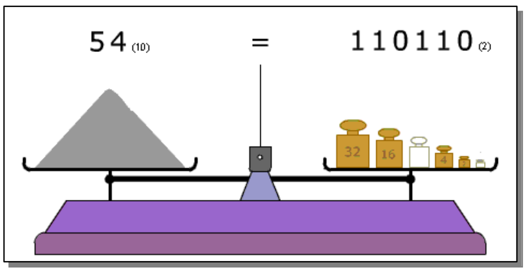
- On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 ⇒ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = B^0 = 1 = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir **1830_{10}** en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : **$111\ 0010\ 0110_2$**

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	1×16^2 = 7 x 256
0		C'est fini, il ne reste plus rien à diviser		

Résultat : **726_{16}** (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

<u>EXERCICES</u>

Rechercher par la méthode de Hornner :

203_8 = ...　　　101010_2 = ...　　　$20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$　　　$COCA_{16}$ = ... $_{10}$

100_{10} = ... $_{2}$　　　1011011_2 = ... $_{10}$

100_{10} = ... $_{16}$　　　236_8 = ... $_{10}$

1023_{10} = ... $_{16}$　　　FFF_{16} = ... $_{10}$

1023_{10} = ... $_{2}$

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 → Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)

- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)

- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)

- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire

- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

□ Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
Cela donne le premier chiffre à gauche, en position p.

Exemple : Soit à convertir $420_{(10)}$ en base 16.

$420_{(10)}$ est supérieur à 16^2 ,
16^2=256 va une fois dans 420
⇒ le chiffre le plus à gauche est 1

Reste à représenter 420 – $1 \text{x} 16^2$ = 164 unités

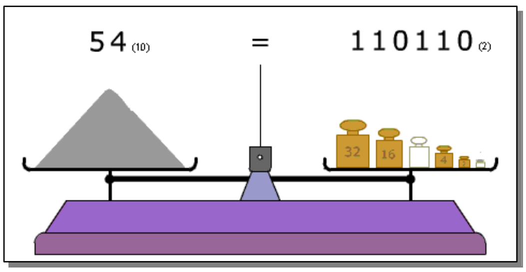
□ On répète la même question tant que le reste est supérieur à la base.

$164_{(10)}$ est supérieur à 16^1 ,
16^1 va 10 fois dans 164 ⇒ chiffre suivant est $A_{(16)}$

Reste 164 – 10x16 = 4 unités

□ Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = $B^0 = 1$ = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste	**0**	0×2^0 = zéro unité
915	: 2	= 457	reste	**1**	1×2^1
457	: 2	= 228	reste	**1**	1×2^2
228	: 2	= 114	reste	**0**	0×2^3
114	: 2	= 57	reste	**0**	0×2^4
57	: 2	= 28	reste	**1**	1×2^5
28	: 2	= 14	reste	**0**	0×2^6
14	: 2	= 7	reste	**0**	0×2^7
7	: 2	= 3	reste	**1**	1×2^8
3	: 2	= 1	reste	**1**	1×2^9
1	: 2	= 0	reste	**1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser			

Le résultat est : $\mathbf{111\ 0010\ 0110_{2}}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste	**6**	6×16^0 = 6 unités
114	: 16	= 7	reste	**2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste	**7**	$1 \times 16^2 = 7 \times 256$
0		C'est fini, il ne reste plus rien à diviser			

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_{2}$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10
« Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
    Multiplier par la base
    Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

<u>EXERCICES</u>

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$

100_{10} = ... $_{2}$

100_{10} = ... $_{16}$

1023_{10} = ... $_{16}$

1023_{10} = ... $_{2}$

$COCA_{16}$ = ... $_{10}$

1011011_2 = ... $_{10}$

236_8 = ... $_{10}$

FFF_{16} = ... $_{10}$

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

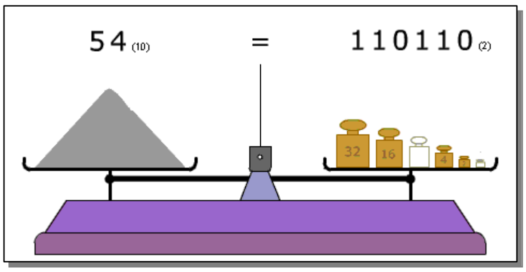
  Reste à représenter 420 – $1\text{x}16^2$ = 164 unités

- On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.



Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.

Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = $B^0 = 1$ = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : $\mathbf{111\ 0010\ 0110_2}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	$1 \times 16^2 = 7 \times 256$
0		C'est fini, il ne reste plus rien à diviser		

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d’une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d’un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l’algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu’il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

EXERCICES

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... 16

100_{10} = ... 2

100_{10} = ... 16

1023_{10} = ... 16

1023_{10} = ... 2

$COCA_{16}$ = ... 10

1011011_2 = ... 10

236_8 = ... 10

FFF_{16} = ... 10

Conversion d’un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d’avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d’un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d’exprimer la même chose est de dire qu’en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l’octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l’hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c’est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d’unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d’unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- Quelle est la plus grande puissance p de la base B que l’on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

  Reste à représenter 420 – 1x16^2 = 164 unités

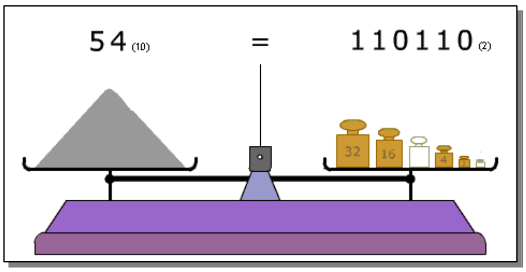
- On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- Le nombre d’unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l’exercice « Pesée d’un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = $B^0 = 1$ = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste **0**	0×2^0 = zéro unité
915	: 2	= 457	reste **1**	1×2^1
457	: 2	= 228	reste **1**	1×2^2
228	: 2	= 114	reste **0**	0×2^3
114	: 2	= 57	reste **0**	0×2^4
57	: 2	= 28	reste **1**	1×2^5
28	: 2	= 14	reste **0**	0×2^6
14	: 2	= 7	reste **0**	0×2^7
7	: 2	= 3	reste **1**	1×2^8
3	: 2	= 1	reste **1**	1×2^9
1	: 2	= 0	reste **1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser		

Le résultat est : $\mathbf{111\ 0010\ 0110_2}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste **6**	6×16^0 = 6 unités
114	: 16	= 7	reste **2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste **7**	$1 \times 16^2 = 7 \times 256$
0		C'est fini, il ne reste plus rien à diviser		

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_2$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
    Multiplier par la base
    Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

<u>EXERCICES</u>

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$

100_{10} = ... $_{2}$

100_{10} = ... $_{16}$

1023_{10} = ... $_{16}$

1023_{10} = ... $_{2}$

$COCA_{16}$ = ... $_{10}$

1011011_2 = ... $_{10}$

236_8 = ... $_{10}$

FFF_{16} = ... $_{10}$

Conversion d'un nombre décimal entier vers une base B quelconque

Base 10 $\rightarrow$ Base B

Rappel : valeur de chaque chiffre

Nous avons vu précédemment comment convertir un nombre de base quelconque en base 10. Il suffit pour ce faire d'avoir compris le principe de la numération de position. Chaque chiffre à une valeur qui dépend du chiffre lui-même et de sa position. On obtient la valeur d'un nombre en additionnant les valeurs des chiffres qui le composent.

Une autre manière d'exprimer la même chose est de dire qu'en lisant un nombre de droite à gauche on rencontre les puissances successives de la base :

- unités, "deuzaines", "quatraines", huitaines, seizaines etc. pour le binaire (base 2)
- unités, huitaines, "soixante-quatraines » etc. pour l'octal (base 8)
- unités, dizaines, centaines, milliers, etc. pour le décimal (base 10)
- unités, seizaines, "deux-cent-cinquante-sixaines", etc. pour l'hexadécimal (base 16)

Voyons à présent comment coder dans une base B quelconque un nombre N dont on connaît la valeur décimale, c'est à dire son écriture en base 10. Il faut pour cela dénombrer les puissances successives de la base :

- le nombre d'unités, de deuzaines, de quatraines etc. pour convertir en binaire
- le nombre d'unités, de seizaines, etc. pour convertir en base 16

Il y a pour ce faire deux méthodes : de "gauche à droite" et de "droite à gauche"

Méthode intuitive : de gauche à droite

- ◻ Quelle est la plus grande puissance p de la base B que l'on puisse retrouver dans N et combien de fois y retrouve-t-on la valeur de B^p ?
  Cela donne le premier chiffre à gauche, en position p.

  Exemple : Soit à convertir $420_{(10)}$ en base 16.

  $420_{(10)}$ est supérieur à 16^2 ,
  16^2=256 va une fois dans 420
  $\Rightarrow$ le chiffre le plus à gauche est 1

  Reste à représenter 420 – 1x16^2 = 164 unités

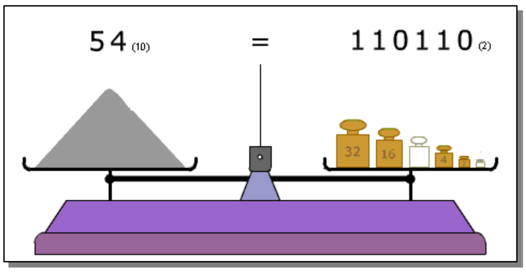
- ◻ On répète la même question tant que le reste est supérieur à la base.

  $164_{(10)}$ est supérieur à 16^1 ,
  16^1 va 10 fois dans 164 $\Rightarrow$ chiffre suivant est $A_{(16)}$

  Reste 164 – 10x16 = 4 unités

- ◻ Le nombre d'unité qui reste inférieur à B est le chiffre le plus à droite autrement dit en position 0 si les positions sont numérotées de droite à gauche.

Retournez au besoin à l'exercice « Pesée d'un nombre binaire » pour vous habituer à évaluer les "poids des bits" qui composent un nombre en base 2.



Méthode systématique : de droite à gauche

Commençons par rechercher la valeur du premier chiffre à droite. Ce chiffre, les unités, est le reste de la division du nombre N à convertir par la base qu'ici nous appellons B. Ce chiffre en position 0 a un poids égal à la base exposant zéro = B^0 = 1 = l'unité.

En divisant à nouveau le quotient de la division précédente par la base on obtient le chiffre de position 1 dont le poids est B^1 = la base.

Des divisions répétées par la base donnent successivement les chiffres de poids B^0, B^1 B^2, B^3, B^4 etc. ce qui nous permet d'écrire le nombre de droite à gauche.

Exemples : Convertir $\mathbf{1830_{10}}$ en binaire ⇒ divisions successives par 2

1830	: 2	= 915	reste	**0**	0×2^0 = zéro unité
915	: 2	= 457	reste	**1**	1×2^1
457	: 2	= 228	reste	**1**	1×2^2
228	: 2	= 114	reste	**0**	0×2^3
114	: 2	= 57	reste	**0**	0×2^4
57	: 2	= 28	reste	**1**	1×2^5
28	: 2	= 14	reste	**0**	0×2^6
14	: 2	= 7	reste	**0**	0×2^7
7	: 2	= 3	reste	**1**	1×2^8
3	: 2	= 1	reste	**1**	1×2^9
1	: 2	= 0	reste	**1**	1×2^{10}
0		C'est fini, il ne reste plus rien à diviser			

Le résultat est : $\mathbf{111\ 0010\ 0110_{2}}$

Convertir **1830** en hexadécimal ⇒ divisions successives par 16

1830	: 16	= 114	reste	**6**	6×16^0 = 6 unités
114	: 16	= 7	reste	**2**	2×16^1 = 2 seizaines
7	: 16	= 0	reste	**7**	$1 \times 16^2 = 7 \times 256$
0		C'est fini, il ne reste plus rien à diviser			

Résultat : $\mathbf{726_{16}}$ (qui concorde bien avec le code $111\ 0010\ 0110_{2}$ trouvé en binaire)

Conclusions :

Ce procédé fonctionne pour toutes les bases mais en informatiques seuls nous concernent le binaire et l'hexadécimal, parfois mais plus rarement l'octal (base 8).

La conversion en binaire est la plus facile, le reste vaut 0 pour les nombres pairs et 1 pour les nombres impairs.

On a donc avantage à convertir d'abord en binaire. Le passage en hexadécimal comme nous l'avons vu au début du cours n'est plus alors qu'un jeu d'enfant.

Autre méthode pour convertir d'une base B en base 10 « Méthode de Horner »

Nous avons vu, au chapitre 2, comment calculer la valeur d'un nombre quelle que soit la base utilisée pour le représenter. Nous additionnions les valeurs obtenues en calculant les valeurs de chaque chiffre compte tenu de leurs positions dans le nombre.

La méthode qui suit donne le même résultat, ceux qui aiment les "recettes de cuisines" la trouveront plus systématique.

Montrons comment cela marche pour le binaire
mais la méthode est valable quelle que soit la base.

Voici l'algorithme :

```
Lire la valeur du chiffre à gauche
Répéter tant qu'il reste des chiffres à droite
{
        Multiplier par la base
        Ajouter le chiffre suivant
}
```

Exemples :

1101_2 = ((((**1** x 2) + **1**) x 2 + **0**) x 2 +**1**

1	**1**	**0**	**1**
	2	6	12
	3	6	13

123_8 = (((**1** x 8) + **2**) x 8) + **3**

1	2	3
	8	80
	10	83

$20C_{16}$ = (((**2** x 16) + **0**) x 16) + **12**

2	**0**	**12**
	32	64
	32	76

EXERCICES

Rechercher par la méthode de Hornner :

203_8 = ... 101010_2 = ... $20A_{16}$ = ...

Méthode au choix :

166_{10} = ... $_{16}$ $COCA_{16}$ = ... $_{10}$

100_{10} = ... $_{2}$ 1011011_2 = ... $_{10}$

100_{10} = ... $_{16}$ 236_8 = ... $_{10}$

1023_{10} = ... $_{16}$ FFF_{16} = ... $_{10}$

1023_{10} = ... $_{2}$