Le problème de la représentation des données

- La mémoire d'un ordinateur est composé de $bits$ pouvant prendre uniquement les valeurs 0 et de 1

Le problème de la représentation des données

- La mémoire d'un ordinateur est composé de $bits$ pouvant prendre uniquement les valeurs 0 et de 1
- Le regroupement de 8 bits s'appelle un octet (*byte* en anglais), c'est l'unité minimal de mémoire :

$$1 \text{ octet } = \underbrace{\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{0}}_{8 \text{ bits}}$$

Le problème de la représentation des données

- La mémoire d'un ordinateur est composé de $bits$ pouvant prendre uniquement les valeurs 0 et de 1
- Le regroupement de 8 bits s'appelle un octet (*byte* en anglais), c'est l'unité minimal de mémoire :

$$1 \text{ octet } = \underbrace{\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{0}}_{8 \text{ bits}}$$

- Toutes les données doivent donc être représentées en utilisant des octets.

Le problème de la représentation des données

- La mémoire d'un ordinateur est composé de $bits$ pouvant prendre uniquement les valeurs 0 et de 1
- Le regroupement de 8 bits s'appelle un octet (*byte* en anglais), c'est l'unité minimal de mémoire :

$$1 \text{ octet } = \underbrace{\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{0}}_{8 \text{ bits}}$$

- Toutes les données doivent donc être représentées en utilisant des octets.
- On s'intéresse ici à la représentation des entiers positifs et négatifs.

De la base 10 à la base 2

- Nous sommes habitués à écrire en utilisant 10 chiffres (0,1,2,3,4,5,6,7,8 et 9), chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.

De la base 10 à la base 2

- Nous sommes habitués à écrire en utilisant 10 chiffres (0,1,2,3,4,5,6,7,8 et 9), chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{1815}^{10}$:

De la base 10 à la base 2

- Nous sommes habitués à écrire en utilisant 10 chiffres (0,1,2,3,4,5,6,7,8 et 9), chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{1815}^{10}$:

  1 8 1 5

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{1815}^{10}$:

10^3	10^2	10^1	10^0
1	8	1	5

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{1815}^{10}$:

10^3	10^2	10^1	10^0
1	8	1	5

$= 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{1815}^{10}$:

10^3	10^2	10^1	10^0
1	8	1	5

$= 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$

- De la même façon, on pourrait utiliser simplement 2 chiffres et multiplier chaque chiffre par une puissance de 2 suivant son emplacement dans le nombre.

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{1815}^{10}$:

10^3	10^2	10^1	10^0
1	8	1	5

$= 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$

- De la même façon, on pourrait utiliser simplement 2 chiffres et multiplier chaque chiffre par une puissance de 2 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{11100010111}^{2}$:

1 1 1 0 0 0 1 0 1 1 1

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour 1815 :

10^3	10^2	10^1	10^0
1	8	1	5

$$= 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

- De la même façon, on pourrait utiliser simplement 2 chiffres et multiplier chaque chiffre par une puissance de 2 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{11100010111}^2$:

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	0	0	1	0	1	1	1

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour 1815 :

10^3	10^2	10^1	10^0
1	8	1	5

$= 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$

- De la même façon, on pourrait utiliser simplement 2 chiffres et multiplier chaque chiffre par une puissance de 2 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{11100010111}^2$:

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	0	0	1	0	1	1	1

$=2^{10}+2^9+2^8+2^4+2^2+2^1+2^0$

De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers postifs en utilisant 10 chiffres, chaque chiffre étant multiplié par une puissance de 10 suivant son emplacement dans le nombre.
  Par exemple, pour 1815 :

10^3	10^2	10^1	10^0
1	8	1	5

$= 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$

- De la même façon, on pourrait utiliser simplement 2 chiffres et multiplier chaque chiffre par une puissance de 2 suivant son emplacement dans le nombre.
  Par exemple, pour $\overline{11100010111}^2$:

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	0	0	1	0	1	1	1

$=2^{10}+2^9+2^8+2^4+2^2+2^1+2^0$

$=1815$

Ce sont des cas particuliers (avec $b = 10$ et $b = 2$), du théorème suivant :

Décomposition en base b

Tout entier $n \in \mathbb{N}$ peut s'écrire sous la forme :

$$n = \sum_{k=0}^{p} a_k b^k$$

avec $p \geq 0$ et $a_k \in [\![0; b-1]\!]$. De plus, cette écriture est unique si $a_p \neq 0$ et s'appelle *décomposition en base b de n* et on la note $n = \overline{a_p \dots a_1 a_0}^b$

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^{2}$

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^2$
- $\overline{1101001011}^2$

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^2$
- $\overline{1101001011}^2$
- $\overline{421}^5$

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^{2}$
- $\overline{1101001011}^{2}$
- $\overline{421}^{5}$
- $\overline{3EA}^{16}$

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^2$
- $\overline{1101001011}^2$
- $\overline{421}^5$
- $\overline{3EA}^{16}$

  On travaille ici en base 16, donc avec 16 chiffres, les lettres majuscules de A à F représentent les "chiffres" 10 à 15.

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^{2} = \overline{139}^{10}$
- $\overline{1101001011}^{2}$
- $\overline{421}^{5}$
- $\overline{3EA}^{16}$

  On travaille ici en base 16, donc avec 16 chiffres, les lettres majuscules de A à F représentent les "chiffres" 10 à 15.

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^{2} = \overline{139}^{10}$
- $\overline{1101001011}^{2} = \overline{843}^{10}$
- $\overline{421}^{5}$
- $\overline{3EA}^{16}$

  On travaille ici en base 16, donc avec 16 chiffres, les lettres majuscules de A à F représentent les "chiffres" 10 à 15.

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^{2} = \overline{139}^{10}$
- $\overline{1101001011}^{2} = \overline{843}^{10}$
- $\overline{421}^{5} = \overline{111}^{10}$
- $\overline{3EA}^{16}$

  On travaille ici en base 16, donc avec 16 chiffres, les lettres majuscules de A à F représentent les "chiffres" 10 à 15.

Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^{2} = \overline{139}^{10}$
- $\overline{1101001011}^{2} = \overline{843}^{10}$
- $\overline{421}^{5} = \overline{111}^{10}$
- $\overline{3EA}^{16} = \overline{1002}^{10}$

  On travaille ici en base 16, donc avec 16 chiffres, les lettres majuscules de A à F représentent les "chiffres" 10 à 15.

Limitations mémoire et dépassement de capacité

- Le nombre de bits représentant un entier est limité, le plus grand nombre représentable sur n bits est :
  $\overline{1\dots1}^2 = 2^{n-1} + \cdots + 1 = 2^n - 1$

Limitations mémoire et dépassement de capacité

- Le nombre de bits représentant un entier est limité, le plus grand nombre représentable sur n bits est :
$\overline{1\dots1}^2 = 2^{n-1} + \cdots + 1 = 2^n - 1$
- Certains langages utilisent un nombre défini de bits pour représenter un entier, on peut donc avoir des problèmes de dépassement de capacités (*overflow*).

Limitations mémoire et dépassement de capacité

- Le nombre de bits représentant un entier est limité, le plus grand nombre représentable sur n bits est :
  $\overline{1\dots1}^2 = 2^{n-1} + \cdots + 1 = 2^n - 1$
- Certains langages utilisent un nombre défini de bits pour représenter un entier, on peut donc avoir des problèmes de dépassement de capacités (*overflow*).
- Python utilise des entiers dit *multi-précision* dont la taille (en nombre de bits) évolue, on a donc pas le problème de dépassement de capacité.

Limitations mémoire et dépassement de capacité

- Le nombre de bits représentant un entier est limité, le plus grand nombre représentable sur n bits est :
$\overline{1\dots 1}^2 = 2^{n-1} + \dots + 1 = 2^n - 1$
- Certains langages utilisent un nombre défini de bits pour représenter un entier, on peut donc avoir des problèmes de dépassement de capacités (*overflow*).
- Python utilise des entiers dit *multi-précision* dont la taille (en nombre de bits) évolue, on a donc pas le problème de dépassement de capacité.
- Par contre, il devient problématique d'évaluer le temps nécessaire à une opération donnée (par exemple une multiplication) sur ces entiers.

Fonction `bin`

La fonction `bin` de Python prend en argument un nombre entier et renvoie la représentation binaire de cet entier sous la forme d'une chaîne de caractères composée de 0 et de 1 et précédée de `"0b"`.

Fonction `bin`

La fonction `bin` de Python prend en argument un nombre entier et renvoie la représentation binaire de cet entier sous la forme d'une chaîne de caractères composée de 0 et de 1 et précédée de `"0b"`.

Exemple

- `bin(10)` renvoie `"0b1010"`
- `bin(255)` renvoie `"0b11111111"`

Complément à deux

- La stratégie qui consiste à prendre *un bit de signe* et à représenter la valeur absolue de l'entier sur les autres présente deux difficultés : 0 est représenté deux fois et surtout l'addition binaire bit à bit ne fonctionne pas.

Complément à deux

- La stratégie qui consiste à prendre *un bit de signe* et à représenter la valeur absolue de l'entier sur les autres présente deux difficultés : 0 est représenté deux fois et surtout l'addition binaire bit à bit ne fonctionne pas.
- La méthode utilisée est celle du complément à 2, sur n bits, on compte négativement le bit de poids 2^{n-1} et positivement les autres.

Complément à deux

- La stratégie qui consiste à prendre *un bit de signe* et à représenter la valeur absolue de l'entier sur les autres présente deux difficultés : 0 est représenté deux fois et surtout l'addition binaire bit à bit ne fonctionne pas.
- La méthode utilisée est celle du complément à 2, sur n bits, on compte négativement le bit de poids 2^{n-1} et positivement les autres.
  Par exemple, sur 8 bits :
  $\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{1}\boxed{0}\boxed{1}\boxed{0} = -2^7 + 2^4 + 2^3 + 2^1 = -101$
- De façon générale, sur n bits, la valeur en complément à deux de la suite bits $(b_{n-1} \dots b_0)$ est :

$$-b_{n-1}\, 2^{n-1} + \sum_{k=0}^{n-2} b_k\, 2^k$$

Conséquences de la représentation en complément à 2

- Les difficultés de la stratégie du *un bit de signe* sont levées.

Conséquences de la représentation en complément à 2

- Les difficultés de la stratégie du *un bit de signe* sont levées.
- Le plus petit petit représentable sur n bits est alors -2^{n-1} et le plus grand $2^{n-1}-1$

Conséquences de la représentation en complément à 2

- Les difficultés de la stratégie du *un bit de signe* sont levées.
- Le plus petit petit représentable sur n bits est alors -2^{n-1} et le plus grand $2^{n-1} - 1$
- Comme pour les entiers positifs, il n'y a pas de problème de dépassement de capacité.

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

Exemples

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

1. on commence par écrire la représentation binaire de la valeur absolue de ce nombre

Exemples

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

1. on commence par écrire la représentation binaire de la valeur absolue de ce nombre
2. on inverse tous les bits de cette représentation

Exemples

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

1. on commence par écrire la représentation binaire de la valeur absolue de ce nombre
2. on inverse tous les bits de cette représentation
3. on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Exemples

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

1. on commence par écrire la représentation binaire de la valeur absolue de ce nombre
2. on inverse tous les bits de cette représentation
3. on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Exemples

1. Quel est le nombre codé en complément à 2 sur 8 bits par $\overline{10110001}^2$?

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

1. on commence par écrire la représentation binaire de la valeur absolue de ce nombre
2. on inverse tous les bits de cette représentation
3. on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Exemples

1. Quel est le nombre codé en complément à 2 sur 8 bits par $\overline{10110001}^2$?
2. Donner l'écriture en complément à 2 sur 8 bits de -12.

Méthode pratique

Pour obtenir la représentation en complément à deux sur n bits d'un entier négatif on pourra utiliser la méthode suivante :

1. on commence par écrire la représentation binaire de la valeur absolue de ce nombre
2. on inverse tous les bits de cette représentation
3. on ajoute 1, sans tenir compte de la dernière retenue éventuelle

Exemples

1. Quel est le nombre codé en complément à 2 sur 8 bits par $\overline{10110001}^2$?
2. Donner l'écriture en complément à 2 sur 8 bits de -12.
3. Donner l'écriture en complément à 2 sur 8 bits de -75.

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits :

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits :

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 :

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.
   1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits :

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8 + 4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.
   1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.
   1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
   2. On inverser tous les bits :

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.
   1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
   2. On inverser tous les bits : 10110100

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.
   1. On écrit $75 = 64+8+2+1$ en binaire sur 8 bits : 01001011
   2. On inverser tous les bits : 10110100
   3. On ajoute 1 :

Correction

1. En complément à 2 sur 8 bits, $\overline{10110001}^2 = -2^7 + 2^5 + 2^4 + 2^0 = -78$
2. Ecriture en complément à 2 sur 8 bits de -12.
   1. On écrit $12 = (8+4)$ en binaire sur 8 bits : 00001100
   2. On inverser tous les bits : 11110011
   3. On ajoute 1 : 11110100
3. Ecriture en complément à 2 sur 8 bits de -75.
   1. On écrit $75 = 64 + 8 + 2 + 1$ en binaire sur 8 bits : 01001011
   2. On inverser tous les bits : 10110100
   3. On ajoute 1 : 10110101

Algorithme des divisions successives

- L'algorithme des divisions successives, permet d'écrire un nombre donnée en base 10 dans n'importe quelle base b. Le principe est d'effectuer les divisions euclidiennes successives par b, les restes de ces divisions sont les chiffres du nombre dans la base b.

Algorithme des divisions successives

- L'algorithme des divisions successives, permet d'écrire un nombre donnée en base 10 dans n'importe quelle base b. Le principe est d'effectuer les divisions euclidiennes successives par b, les restes de ces divisions sont les chiffres du nombre dans la base b.
- Pour écrire N en base b :

Algorithme des divisions successives

- L'algorithme des divisions successives, permet d'écrire un nombre donnée en base 10 dans n'importe quelle base b. Le principe est d'effectuer les divisions euclidiennes successives par b, les restes de ces divisions sont les chiffres du nombre dans la base b.
- Pour écrire N en base b :
  1. Faire la division euclidienne de N par b, soit Q le quotient et R le reste. (c'est à dire écrire $N = Q \times b + R$ avec $R < b$)

Algorithme des divisions successives

- L'algorithme des divisions successives, permet d'écrire un nombre donnée en base 10 dans n'importe quelle base b. Le principe est d'effectuer les divisions euclidiennes successives par b, les restes de ces divisions sont les chiffres du nombre dans la base b.
- Pour écrire N en base b :
  1. Faire la division euclidienne de N par b, soit Q le quotient et R le reste. (c'est à dire écrire $N = Q \times b + R$ avec $R < b$)
  2. Ajouter R aux chiffres de N en base b

Algorithme des divisions successives

- L'algorithme des divisions successives, permet d'écrire un nombre donnée en base 10 dans n'importe quelle base b. Le principe est d'effectuer les divisions euclidiennes successives par b, les restes de ces divisions sont les chiffres du nombre dans la base b.
- Pour écrire N en base b :
  1. Faire la division euclidienne de N par b, soit Q le quotient et R le reste. (c'est à dire écrire $N = Q \times b + R$ avec $R < b$)
  2. Ajouter R aux chiffres de N en base b
  3. Si $Q = 0$ s'arrêter, sinon recommencer à partir de l'étape 1 en remplaçant N par Q.

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$2019 \quad = \qquad \times \quad 16 \quad +$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$2019 = 126 \times 16 +$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$2019 = 126 \times 16 + \boxed{3}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll} 2019 & = & 126 & \times & 16 & + & \boxed{3} \\ 126 & = & & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll} 2019 & = & 126 & \times & 16 & + & \boxed{3} \\ 126 & = & 7 & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$
\begin{array}{lllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14}
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$
\begin{array}{lllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14} \\
7 & = & & \times & 16 & + &
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{llllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14} \\
7 & = & 0 & \times & 16 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14} \\
7 & = & 0 & \times & 16 & + & \boxed{7}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14} \\
7 & = & 0 & \times & 16 & + & \boxed{7}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14} \\
7 & = & 0 & \times & 16 & + & \boxed{7}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll}
2019 & = & 126 & \times & 16 & + & \boxed{3} \\
126 & = & 7 & \times & 16 & + & \boxed{14} \\
7 & = & 0 & \times & 16 & + & \boxed{7}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{2019}^{10}$.

$$\begin{array}{lllllll} 2019 & = & {\color{blue}126} & \times & 16 & + & {\color{red}\boxed{3}} \\ {\color{blue}126} & = & {\color{blue}7} & \times & 16 & + & {\color{red}\boxed{14}} \\ {\color{blue}7} & = & {\color{blue}0} & \times & 16 & + & {\color{red}\boxed{7}} \end{array}$$

Le quotient est nul, l'algorithme s'arrête et les chiffres en base 16 sont les restes obtenus à chaque étape donc $\overline{2019}^{10} = \overline{7E3}^{16}$ (car 14 correspond au chiffre E).

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$9787 \quad = \qquad\qquad \times \quad 16 \quad +$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$9787 \quad = \quad 611 \quad \times \quad 16 \quad +$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$9787 \quad = \quad {\color{blue}611} \quad \times \quad 16 \quad + \quad \boxed{{\color{red}11}}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & 38 & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & 38 & \times & 16 & + & \boxed{3} \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & 38 & \times & 16 & + & \boxed{3} \\ 38 & = & & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & 38 & \times & 16 & + & \boxed{3} \\ 38 & = & 2 & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lcllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & 38 & \times & 16 & + & \boxed{3} \\ 38 & = & 2 & \times & 16 & + & \boxed{6} \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll}
9787 & = & 611 & \times & 16 & + & \boxed{11} \\
611 & = & 38 & \times & 16 & + & \boxed{3} \\
38 & = & 2 & \times & 16 & + & \boxed{6} \\
2 & = & & \times & 16 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll} 9787 & = & 611 & \times & 16 & + & \boxed{11} \\ 611 & = & 38 & \times & 16 & + & \boxed{3} \\ 38 & = & 2 & \times & 16 & + & \boxed{6} \\ 2 & = & 0 & \times & 16 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll}
9787 & = & 611 & \times & 16 & + & \boxed{11} \\
611 & = & 38 & \times & 16 & + & \boxed{3} \\
38 & = & 2 & \times & 16 & + & \boxed{6} \\
2 & = & 0 & \times & 16 & + & \boxed{2}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 16 de $\overline{9787}^{10}$.

$$\begin{array}{lllllll}
9787 & = & 611 & \times & 16 & + & \boxed{11} \\
611 & = & 38 & \times & 16 & + & \boxed{3} \\
38 & = & 2 & \times & 16 & + & \boxed{6} \\
2 & = & 0 & \times & 16 & + & \boxed{2}
\end{array}$$

Le quotient est nul, l'algorithme s'arrête et les chiffres en base 16 sont les restes obtenus à chaque étape donc $\overline{9781}^{10} = \overline{263B}^{16}$ (car 11 correspond au chiffre B).

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$786 = \quad \times \quad 2 \quad +$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$786 = 393 \times 2 +$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$786 = 393 \times 2 + \boxed{0}$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{ccccccc} 786 & = & 393 & \times & 2 & + & \boxed{0} \\ 393 & = & & \times & 2 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{ccccccc} 786 & = & 393 & \times & 2 & + & \boxed{0} \\ 393 & = & 196 & \times & 2 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{ccccccc} 786 & = & 393 & \times & 2 & + & \boxed{0} \\ 393 & = & 196 & \times & 2 & + & \boxed{1} \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{ccccccc} 786 & = & 393 & \times & 2 & + & \boxed{0} \\ 393 & = & 196 & \times & 2 & + & \boxed{1} \\ 196 & = & & \times & 2 & + & \end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{ccccccc}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{ccccccc}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$
\begin{array}{ccccccc}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + &
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$
\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + &
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$
\begin{array}{llllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & & \times & 2 & + &
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$
\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & & \times & 2 & + &
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lclclcl}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & 1 & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$
\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & 1 & \times & 2 & + & \boxed{1}
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lcllllc}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & 1 & \times & 2 & + & \boxed{1} \\
1 & = & & \times & 2 & + &
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$
\begin{array}{lllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & 1 & \times & 2 & + & \boxed{1} \\
1 & = & 0 & \times & 2 & + &
\end{array}
$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{lclcccc}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & 1 & \times & 2 & + & \boxed{1} \\
1 & = & 0 & \times & 2 & + & \boxed{1}
\end{array}$$

Exemple d'utilisation de l'algorithme des divisions successives

Donner l'écriture en base 2 de $\overline{786}^{10}$.

$$\begin{array}{llllllll}
786 & = & 393 & \times & 2 & + & \boxed{0} \\
393 & = & 196 & \times & 2 & + & \boxed{1} \\
196 & = & 98 & \times & 2 & + & \boxed{0} \\
98 & = & 49 & \times & 2 & + & \boxed{0} \\
49 & = & 24 & \times & 2 & + & \boxed{1} \\
24 & = & 12 & \times & 2 & + & \boxed{0} \\
12 & = & 6 & \times & 2 & + & \boxed{0} \\
6 & = & 3 & \times & 2 & + & \boxed{0} \\
3 & = & 1 & \times & 2 & + & \boxed{1} \\
1 & = & 0 & \times & 2 & + & \boxed{1}
\end{array}$$

Le quotient est nul, l'algorithme s'arrête et $\overline{786}^{10} = \overline{1100010010}^{2}$.

Algorithme en pseudo-code

Algorithme : Conversion de la base 10 vers la base b

Entrées : $n \in \mathbb{N}$ (en base 10) et $b \in \mathbb{N}, b \geqslant 2$.
Sorties : Les chiffres $a_{p-1}, \dots a_0$ de n en base b (donc des éléments de $[\![0; b-1]\!]$)

```
si n = 0 alors
    return 0
fin
p ← nombre de chiffres de n en base b
pour i ← 0 à p − 1 faire
    a_i ← reste dans la division euclidienne de n par b
    n ← ⌊n/b⌋
fin
return a_{p−1}, ..., a_0
```

Algorithme en pseudo-code

Algorithme : Conversion de la base 10 vers la base b

Entrées : $n \in \mathbb{N}$ (en base 10) et $b \in \mathbb{N}, b \geqslant 2$.
Sorties : Les chiffres $a_{p-1}, \dots a_0$ de n en base b (donc des éléments de $[\![0; b-1]\!]$)

```
si n = 0 alors
    return 0
fin
p ← nombre de chiffres de n en base b
pour i ← 0 à p − 1 faire
    a_i ← reste dans la division euclidienne de n par b
    n ← ⌊n / b⌋
fin
return a_{p−1}, ..., a_0
```

Une implémentation sera vue en TP.