

## 1. Introduction

### 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

# C9 Représentation des entiers

## 1. Introduction

### Le problème de la représentation des données

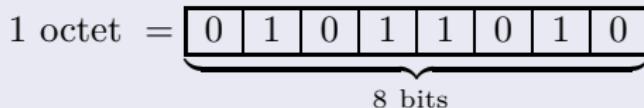
- La mémoire d'un ordinateur est composé de *bits* pouvant prendre uniquement les valeurs 0 et de 1
- Le regroupement de 8 bits s'appelle un octet (*byte* en anglais), c'est l'unité minimal de mémoire :

1 octet =   
8 bits

## 1. Introduction

### 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 :



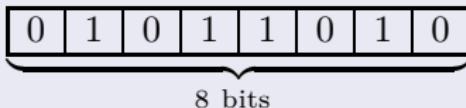
- Toutes les données doivent donc être **représentées** en utilisant des octets.

# C9 Représentation des entiers

## 1. Introduction

### 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 octet =  8 bits

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

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### 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  $\underline{1815}^{10}$  :

# C9 Représentation des entiers

## 2. Entiers positifs

### 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  $\underline{1815}^{10}$  :

1      8      1      5

# C9 Représentation des entiers

## 2. Entiers positifs

### De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers positifs en utilisant **10** chiffres, chaque chiffre étant multiplié par une puissance de **10** suivant son emplacement dans le nombre.

Par exemple, pour  $\underline{1815}^{10}$  :

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

# C9 Représentation des entiers

## 2. Entiers positifs

### 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  $\underline{1815}^{10}$  :

$$\begin{array}{c|c|c|c} 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1 & 8 & 1 & 5 \end{array} = 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

# C9 Représentation des entiers

## 2. Entiers positifs

### De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers positifs en utilisant **10** chiffres, chaque chiffre étant multiplié par une puissance de **10** suivant son emplacement dans le nombre.

Par exemple, pour  $\underline{1815}^{10}$  :

$$\begin{array}{c|c|c|c} 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1 & 8 & 1 & 5 \end{array} = 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

- 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers positifs en utilisant **10** chiffres, chaque chiffre étant multiplié par une puissance de **10** suivant son emplacement dans le nombre.

Par exemple, pour  $\underline{1815}^{10}$  :

$$\begin{array}{c|c|c|c} 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1 & 8 & 1 & 5 \end{array} = 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

- 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  $\underline{11100010111}^2$  :

1    1    1    0    0    0    1    0    1    1    1

## C9 Représentation des entiers

### 2. Entiers positifs

#### De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers positifs en utilisant **10** chiffres, chaque chiffre étant multiplié par une puissance de **10** suivant son emplacement dans le nombre.

Par exemple, pour **1815** :

$$\begin{array}{c|c|c|c} 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1 & 8 & 1 & 5 \end{array} = 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

- 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 **11100010111<sup>2</sup>** :

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c} 2^{10} & 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$

## C9 Représentation des entiers

### 2. Entiers positifs

#### De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers positifs en utilisant **10** chiffres, chaque chiffre étant multiplié par une puissance de **10** suivant son emplacement dans le nombre.

Par exemple, pour **1815** :

$$\begin{array}{c|c|c|c} 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1 & 8 & 1 & 5 \end{array} = 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

- 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 **11100010111**<sup>2</sup> :

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c} 2^{10} & 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$
$$= 2^{10} + 2^9 + 2^8 + 2^4 + 2^2 + 2^1 + 2^0$$

## C9 Représentation des entiers

### 2. Entiers positifs

#### De la base 10 à la base 2

- Nous sommes habitués à écrire les entiers positifs en utilisant **10** chiffres, chaque chiffre étant multiplié par une puissance de **10** suivant son emplacement dans le nombre.

Par exemple, pour **1815** :

$$\begin{array}{c|c|c|c} 10^3 & 10^2 & 10^1 & 10^0 \\ \hline 1 & 8 & 1 & 5 \end{array} = 1 \times 1000 + 8 \times 100 + 1 \times 10 + 5 \times 1 = 1815$$

- 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 **11100010111**<sup>2</sup> :

$$\begin{array}{c|c|c|c|c|c|c|c|c|c|c} 2^{10} & 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array}$$
$$= 2^{10} + 2^9 + 2^8 + 2^4 + 2^2 + 2^1 + 2^0$$
$$= 1815$$

## C9 Représentation des entiers

### 2. Entiers positifs

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

# C9 Représentation des entiers

## 2. Entiers positifs

### Exemples

Ecrire en base 10 les nombres ci-dessous

- $\overline{10001011}^2$

# C9 Représentation des entiers

## 2. Entiers positifs

### Exemples

Ecrire en base 10 les nombres ci-dessous

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

# C9 Représentation des entiers

## 2. Entiers positifs

### Exemples

Ecrire en base 10 les nombres ci-dessous

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

# C9 Représentation des entiers

## 2. Entiers positifs

### Exemples

Ecrire en base 10 les nombres ci-dessous

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

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### Limitations mémoire et dépassement de capacité

- Le nombre de bits représentant un entier est limité, le plus grand nombre représentable sur  $n$  bits est :

$$\overline{1 \dots 1}^2 = 2^{n-1} + \dots + 1 = 2^n - 1$$

# C9 Représentation des entiers

## 2. Entiers positifs

### 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*).

## C9 Représentation des entiers

### 2. Entiers positifs

#### Limitations mémoire et dépassement de capacité

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

# C9 Représentation des entiers

## 2. Entiers positifs

### 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.

# C9 Représentation des entiers

## 2. Entiers positifs

### 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".

## 2. Entiers positifs

### 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"

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### 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.

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### 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.

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 :

$$\boxed{1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0} = -2^7 + 2^4 + 2^3 + 2^1 = -101$$

- De façon générale, sur  $n$  bits, la valeur en **complément à deux** de la suite bits  $(b_{n-1} \dots b_0)$  est :

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

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

Conséquences de la représentation en complément à 2

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Conséquences de la représentation en complément à 2

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

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### Conséquences de la représentation en complément à 2

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Méthode pratique

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

### Exemples

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### Méthode pratique

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

- ❶ on commence par écrire la représentation binaire de la valeur absolue de ce nombre

#### Exemples

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### Méthode pratique

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

- ❶ on commence par écrire la représentation binaire de la valeur absolue de ce nombre
- ❷ on inverse tous les bits de cette représentation

#### Exemples

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### 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 10110001<sup>2</sup> ?

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### 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 10110001<sup>2</sup> ?
- ② Donner l'écriture en complément à 2 sur 8 bits de -12.

## C9 Représentation des entiers

### 3. Représentation des entiers négatifs

#### 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 10110001<sup>2</sup> ?
- 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.

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Correction

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 :

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 :

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 inverse tous les bits : 11110011
  3. On ajoute 1 :

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 inverse tous les bits : 11110011
  3. On ajoute 1 : 11110100

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Correction

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 inverse 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 :

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Correction

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Correction

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### Correction

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

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 inverse 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 inverse tous les bits : 10110100
  3. On ajoute 1 :

# C9 Représentation des entiers

## 3. Représentation des entiers négatifs

### 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 inverse 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 inverse tous les bits : 10110100
  3. On ajoute 1 : 10110101

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### 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  $b$ .

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### 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  $b$  :

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### 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  $b$  :
  - ➊ 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  $R < b$ )

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### 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  $b$  :
  - ➊ 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  $R < b$ )
  - ➋ Ajouter  $R$  aux chiffres de  $N$  en base  $b$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### 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  $b$  :
  - ➊ 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  $R < b$ )
  - ➋ Ajouter  $R$  aux chiffres de  $N$  en base  $b$
  - ➌ Si  $Q = 0$  s'arrêter, sinon recommencer à partir de l'étape 1 en remplaçant  $N$  par  $Q$ .

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

Exemple d'utilisation de l'algorithme des divisions successives

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

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 + 3$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 + 3$$

$$126 = \quad \times 16 +$$

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 + 3$$

$$126 = 7 \times 16 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 + 3$$

$$126 = 7 \times 16 + 14$$

$$7 = 0 \times 16 + 7$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 + 3$$

$$126 = 7 \times 16 + 14$$

$$7 = 0 \times 16 + 7$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$2019 = 126 \times 16 + 3$$

$$126 = 7 \times 16 + 14$$

$$7 = 0 \times 16 + 7$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

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).

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

Exemple d'utilisation de l'algorithme des divisions successives

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$9787 = 611 \times 16 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$9787 = 611 \times 16 + 11$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$\begin{array}{rcl} 9787 & = & 611 \times 16 + 11 \\ 611 & = & \quad \quad \times 16 + \end{array}$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$9787 = 611 \times 16 + 11$$

$$611 = 38 \times 16 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

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).

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$786 = 393 \times 2 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$786 = 393 \times 2 + 0$$

$$393 = \quad \times 2 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

$$786 = 393 \times 2 + 0$$

$$393 = 196 \times 2 +$$

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

## C9 Représentation des entiers

### 4. Algorithme des divisions successives

#### Exemple d'utilisation de l'algorithme des divisions successives

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

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

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

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Algorithme en pseudo-code

#### Algorithme : Conversion de la base 10 vers la base b

**Entrées :**  $n \in \mathbb{N}$  (en base 10) et  $b \in \mathbb{N}, b \geq 2$ .

**Sorties :** Les chiffres  $a_{p-1}, \dots, a_0$  de  $n$  en base  $b$  (donc des éléments de  $\llbracket 0; b - 1 \rrbracket$ )

```
1 si  $n = 0$  alors
2   | return 0
3 fin
4  $p \leftarrow$  nombre de chiffres de  $n$  en base  $b$ 
5 pour  $i \leftarrow 0$  à  $p - 1$  faire
6   |  $a_i \leftarrow$  reste dans la division euclidienne de  $n$  par  $b$ 
7   |  $n \leftarrow \lfloor \frac{n}{b} \rfloor$ 
8 fin
9 return  $a_{p-1}, \dots, a_0$ 
```

# C9 Représentation des entiers

## 4. Algorithme des divisions successives

### Algorithme en pseudo-code

#### Algorithme : Conversion de la base 10 vers la base b

**Entrées :**  $n \in \mathbb{N}$  (en base 10) et  $b \in \mathbb{N}, b \geq 2$ .

**Sorties :** Les chiffres  $a_{p-1}, \dots, a_0$  de  $n$  en base  $b$  (donc des éléments de  $\llbracket 0; b - 1 \rrbracket$ )

```
1 si  $n = 0$  alors
2   | return 0
3 fin
4  $p \leftarrow$  nombre de chiffres de  $n$  en base  $b$ 
5 pour  $i \leftarrow 0$  à  $p - 1$  faire
6   |  $a_i \leftarrow$  reste dans la division euclidienne de  $n$  par  $b$ 
7   |  $n \leftarrow \lfloor \frac{n}{b} \rfloor$ 
8 fin
9 return  $a_{p-1}, \dots, a_0$ 
```

Une implémentation sera vue en TP.