

Représentation des entiers positifs

Exercices

Dernière mise à jour le : 17/08/2023

■ Partie 1 : Algorithmes de conversion

Conversion d'un entier en base 2

Il s'agit d'écrire ici l'algorithme en pseudo-code de la conversion en binaire par suite de divisions. On dit aussi qu'il s'agit de la méthode euclidienne (car on fait des divisions euclidiennes). Vous avez déjà déroulé cet algorithme sur des exemples et vous êtes rendus compte que pour obtenir l'écriture binaire il fallait concaténer les restes des divisions euclidiennes.

 **Question 1 :** Quel type de données de base permet de concaténer des éléments ?

 **Question 2 :** L'algorithme consiste à répéter plusieurs fois les mêmes instructions et s'arrête lorsqu'on obtient un quotient nul. Quel type de structure vous paraît adapter pour écrire cet algorithme ? Pourquoi ?

 **Question 3 : Écriture de l'algorithme.**

Complétez l'algorithme suivant de manière à ce qu'il corresponde à la spécification suivante

Spécification :

- Entrées : un entier positif `n`
- Sortie : une chaîne de caractères `D`
- Rôle : convertir `n` en base 2
- Précondition : aucune
- Postcondition : `D` représente l'écriture en base 2 de `n`

Algorithme :

```
D ← ..... # valeur de D avant la première division
tant que n ..... faire
    r ← ..... # reste de la division euclidienne
    D ← concaténation( ... , ... )
    ... ← n div 2 # quotient de la division
fin tant que
```

 **Question 4 :** Programmez cet algorithme en Python et testez-le sur des exemples, en comparant avec la fonction `bin()` de Python..

 **Question 5 :** Transformez ce script Python en une fonction, appelée `entier_en_binaire` qui prend en paramètre l'entier `n` et qui renvoie la chaîne `D`. Vous pourrez ainsi, réutiliser cette fonction !

Exemples :

```
>>> entier_en_binaire(5)
'101'
>>> binaire_en_entier(255)
'11111111'
```

Conversion binaire vers décimale

Le but est désormais d'écrire l'algorithme de la conversion inverse. Cet algorithme répond à la spécification suivante :

Spécification :

- Entrées : une chaîne de caractères `D`
- Sortie : un entier `n`
- Rôle : convertir `D` en base 10
- Précondition : `D` est formé uniquement de caractères 0 et 1
- Postcondition : `n` est l'écriture en base 10 de `D`

 **Question 6** : Quel type de structure permet de parcourir une chaîne de caractères très rapidement ?

 **Question 7 : Écriture de l'algorithme**

Compléter l'algorithme suivant de manière à ce qu'il corresponde à la spécification précédente.

Attention : lorsque l'on parcourt la chaîne `D` avec une boucle `for`, on commence par le chiffre de gauche et donc par la plus grande puissance de 2. Il faut aussi penser à convertir les caractères en des entiers pour faire les calculs.

```
n ← .....          # valeur de n avant les calculs
i ← .....          # plus grande puissance de 2 au départ
pour carac dans D faire
    n ← n + .....   # ajout du chiffre multiplié par sa puissance de 2
    ... ← .....     # la puissance diminue de 1
fin pour
```

 **Question 8** : Codez une fonction Python, appelée `binaire_en_entier` qui prend en paramètre une chaîne de caractères `D` (constituée de `0` et de `1`) et qui renvoie l'entier `n`. Vous pourrez ainsi, réutiliser cette fonction.

Exemples :

```
>>> binaire_en_entier('101')
5
>>> binaire_en_entier('11111111')
255
```

 **Question 9** (optionnelle) : Programmez des fonctions Python permettant de coder les conversions "décimal vers hexadécimal" et "hexadécimal vers décimal". La difficulté réside dans les conversions entre les caractères `'A'`, `'B'`, ..., `'F'` et leurs valeurs entières.

■ Partie 2 : Ajouter et multiplier des entiers naturels en base 2

On peut poser les additions et multiplications comme en base 10 !

Exemple 1 : Calcul de $(101)_2 + (1101)_2$

On pose l'addition :

$$\begin{array}{r} 1\ 1 \quad \leftarrow \text{retenues} \\ 101 \\ + 1101 \\ \hline 10010 \quad \leftarrow \text{résultat} \end{array}$$

Vérification : $(101)_2 + (1101)_2 = 5 + 13 = 18 = (10010)_2$.

Exemple 2 : Calcul de $(1110)_2 \times (101)_2$

On pose la multiplication :

$$\begin{array}{r} 1110 \\ \times 101 \\ \hline 11 \quad \leftarrow \text{retenues pour l'addition} \\ 1110 \\ + 0000. \\ + 1110.. \\ \hline 1000110 \quad \leftarrow \text{résultat} \end{array}$$

Vérification : $(1110)_2 \times (101)_2 = 14 \times 5 = 70 = (100110)_2$.

 **Question** : Calculer la somme $(1011)_2 + (11)_2$ et le produit $(110)_2 \times (100)_2$.

Vous vérifierez vos résultats en repassant par la base décimale.

■ Partie 3 : Nombre de bits nécessaires à une écriture binaire

Cas de la somme : Ajouter deux entiers naturels ayant respectivement k et ℓ bits significatifs en nécessite au plus $\max(k, \ell) + 1$.

Par exemple, si on additionne un entier de 3 bits avec un autre de 5 bits, alors la somme s'écrit au plus sur 6 bits.

Cas du produit : Multiplier deux entiers naturels n et m ayant respectivement k et ℓ bits significatifs en nécessite au plus $k + \ell$.

Par exemple, si on multiplie un entier de 3 bits par un autre de 5 bits, alors le produit s'écrit au plus sur 8 bits.

Exercice 1

Dans tout l'exercice, les nombres n et m désignent des entiers naturels ayant respectivement 4 et 5 bits significatifs.

1. Quel est le nombre maximal de bits significatifs nécessaires pour coder l'entier $n + m$?
Et l'entier $n \times m$?
2. Quelles sont les valeurs binaires minimales et maximales pour n et m ?
3. Donnez deux nombres binaires n et m tels que l'écriture binaire de $n + m$ nécessite 6 bits.
Même question pour 5 bits.
4. Supposons que $n = (1111)_2$.
Peut-on trouver un nombre m tel que l'écriture binaire de $n + m$ nécessite 5 bits ?
5. Supposons que $m = (11000)_2$.
Peut-on trouver un nombre n tel que l'écriture binaire de $n + m$ nécessite 5 bits ?
6. Supposons que $m = (10101)_2$.
Quel est le nombre minimal de bits pour l'écriture binaire de $n \times m$?

Exercice 2

Déterminez dans chaque cas le nombre de bits nécessaires à l'écriture en base 2 de la somme $n + m$ puis du produit $n \times m$.

1. $n = 12$ et $m = 7$.
2. $n = (1011)_2$ et $m = (11100)_2$.

Références :

- Synthèse de cours NSI 2019-2020, Mickaël Barraud

Germain Becker & Sébastien Point, Lycée Emmanuel Mounier, Angers



Voir en ligne : info-mounier.fr/premiere_nsi/types_base/entiers-positifs-exercices