

# Fifi

1<sup>er</sup> février 2018

## 1 le background

Le codage de Fibonacci est un codage qui utilise les nombres de la suite de Fibonacci pour encoder les nombres entiers positifs en une suite binaire.

L'idée est d'utiliser le théorème de Zeckendorf qui dit que tout nombre entier peut se décomposer de manière unique comme la somme de terme non consécutifs de la suite de Fibonacci.

## 2 l'énoncé

L'objectif est de coder des entiers positifs en utilisant le codage de Fibonacci. Pour cela, votre programme doit lire sur l'entrée standard :

1. un entier  $N$  représentant le nombre de valeurs à coder ;
2.  $N$  lignes donnant les valeurs.

Votre programme doit écrire sur la sortie standard  $N$  lignes. La  $i^e$  ligne donne la représentation dans le codage de Fibonacci de la  $i^e$  valeur d'entrée.

Un nombre  $X$  est codé dans le codage de Fibonacci par une suite de bits  $b_0b_1b_2 \dots b_k$  telle que  $b_{k-1} = b_k = 1$  et

$$X = \sum_{i=0}^{k-1} b_i F(i-2)$$

où  $F(n)$  est le  $n^e$  terme de la suite de Fibonacci.

Pour rappel, la suite de Fibonacci  $F(n)$  est définie de la manière suivante : chaque terme est la somme des deux précédents.

$$F(n) = \begin{cases} 0 & \text{pour } n = 0 \\ 1 & \text{pour } n = 1 \\ F(n) = F(n-2) + F(n-1) & \text{pour } n > 1 \end{cases}$$

Ainsi  $F(2) = F(0) + F(1) = 1$ ,  $F(3) = F(1) + F(2) = 2$ ,  $F(4) = F(2) + F(3) = 3$ , etc.

Pour encoder un nombre  $X$ , on pourra utiliser la procédure suivante :

1. le dernier bit est un 1 (toujours) ;
2. trouver le plus grand nombre de Fibonacci  $F(i)$  inférieur ou égal à  $X$  ;
3. si  $F(i) \leq X$ , retrancher  $F(i)$  à  $X$ , et ajouter un 1 au début du code ;
4. sinon, ajouter un 0 au début du code ;
5. tant que  $i > 1$ , passer au nombre de Fibonacci immédiatement inférieur, et recommencer au point 3.

Un exemple d'entrée/sortie est donné dans le tableau 1.

entrée	sortie	décomposition de Fibonacci
15		
1	11	$F(2) = 1$
2	011	$F(3) = 2$
3	0011	$F(4) = 3$
4	1011	$F(2) + F(4) = 1 + 3$
5	00011	$F(5) = 5$
6	10011	$F(2) + F(5) = 1 + 5$
7	01011	$F(3) + F(5) = 2 + 5$
8	000011	$F(6) = 8$
9	100011	$F(2) + F(6) = 1 + 8$
10	010011	$F(3) + F(6) = 2 + 8$
11	001011	$F(4) + F(6) = 3 + 8$
12	101011	$F(2) + F(3) + F(6) = 1 + 3 + 8$
13	0000011	$F(7) = 13$
14	1000011	$F(2) + F(7) = 1 + 13$
15	0100011	$F(3) + F(7) = 2 + 13$

TABLE 1 – Exemple d'entrée et la sortie attendue

### Contraintes

- le nombre  $N$  de valeurs à coder est compris entre 1 et  $2^{20}$  inclus ;
- les valeurs à coder sont comprises entre 1 et  $2^{30}$  inclus.

## 3 les ressources

Pour vous aider dans la réalisation du programme, vous trouverez sur

<http://cours-info.iut-bm.univ-fcomte.fr>

un article dans la section `hackaton` → édition 2018, portant le même titre que l'exercice. Il contient un lien permettant de télécharger le fichier d'entrée donné ci-dessus.

Bien entendu, vous êtes libres d'utiliser ou non ce canevas, mais c'est un gain de temps que de s'en servir comme base.