Bases complètes

Une suite d'entiers est dite complète si tout entier positif peut être représenté par la somme d'un sous-ensemble de la suite. Une telle suite bien connue est la suite des puissances de 2, c'est-à-dire {1, 2, 4, 8, 16, …}, c'est pourquoi tout nombre peut être représenté en binaire. D'autres suites bien connues ayant cette propriété comprennent la suite de Fibonacci ainsi que le nombre 1 suivi de l'ensemble des nombres premiers.

En binaire, un 0 ou un 1 est utilisé pour indiquer si l'élément de la suite des puissances de 2 correspondant est inclus dans la somme. Suivant ce même principe, il est possible de représenter les entiers en utilisant une autre suite complète.

Dans la suite de Fibonacci et celle avec les nombres premiers, il existe plusieurs représentations différentes d'un même nombre. Par exemple, le nombre 18 peut être obtenu avec 13 + 5 ou 13 + 3 + 2. Étonnamment, il est possible d'éviter ces doublons avec la suite de Fibonacci. Pour ce faire, il faut éliminer le premier 1 dans la suite (puisque la valeur suivante est aussi un 1) ainsi qu'éviter d'utiliser deux valeurs consécutives de la suite (puisque la valeur suivante est la somme des deux valeurs). En ajoutant ces deux contraintes, il est possible de représenter tout les nombres positifs de façon unique.

Le tableau suivant montre quelques exemples. La représentation avec la suite de Fibonacci est la représentation unique attendue.

Décimal	Binaire 128, 64, 32, 16, 8, 4, 2, 1	Nombres premiers 17, 13, 11, 7, 5, 3, 2, 1	Fibonacci 34, 21, 13, 8, 5, 3, 2, 1
10	0000 1010	0001 0100	0001 0010
20	0001 0100	0011 0010	0010 1010
50	0011 0010	1110 1101	1010 0100

Le problème qui vous est posé consiste à jongler avec ces différentes bases. Vous recevrez en entrée un nombre écrit en décimal et devrez suivre les étapes suivantes :

1. Représenter le nombre en binaire afin d'avoir une séquence de 0 et 1 avec laquelle travailler.
2. Utiliser la séquence obtenue et calculer le nombre qu'elle représente dans la base des nombres premiers.
3. Retourner la représentation dans la base Fibonacci du nombre obtenu.

Par exemple, pour le nombre 50, voici le détail des calculs :

1. $50 = 00110010_2$
2. $00110010_{\text{Premiers}} = 11 + 7 + 2 = 20$
3. $20 = 13 + 5 + 2 = 00101010_{\text{Fibonacci}}$

La valeur à retourner est donc 101010

Entrée : Un entier positif en décimal.

Sortie : La représentation unique en base Fibonacci obtenue. Il faut enlever les 0 au début puisqu'ils sont inutiles.

Exemples :

Entrée	Sortie
50	101010
1024	1010000
2047	1010001010
1000000	100101010010