

Soustraction

1 Descriptif

L'objectif de ce défi est d'implémenter la soustraction de deux nombres binaires. On supposera toujours que le premier nombre est plus grand que le deuxième. L'idée générale de cet algorithme est de manipuler en permanence trois variables :

- Une retenue R (initialisée à 0)
- Le bit du premier nombre B_1
- Le bit du second nombre B_2

En regardant le résultat de $B_1 - B_2 - R$, il est possible d'en déduire le bit du résultat et la nouvelle valeur de la retenue.

Par exemple, si on essaye de faire la soustraction suivante : 1001 - 111, l'algorithme se déroule de la façon suivante :

1. On prend le premier bit des deux nombres : $B_1 = 1$, $B_2 = 1$ et $R = 0$.
2. Comme $B_1 - B_2 - R = 0$, on en déduit que le premier bit du résultat est 0 et R prend la valeur 0.
3. On prend le deuxième bit des deux nombres : $B_1 = 0$, $B_2 = 1$ et $R = 0$.
4. Comme $B_1 - B_2 - R = -1$, on en déduit que le deuxième bit du résultat est 1 et R prend la valeur 1.
5. On prend le troisième bit des deux nombres : $B_1 = 0$, $B_2 = 1$ et $R = 1$.
6. Comme $B_1 - B_2 - R = -2$, on en déduit que le troisième bit du résultat est 0 et R prend la valeur 1.
7. On prend le quatrième bit des deux nombres : $B_1 = 1$, $B_2 = 0$ et $R = 1$.
8. Comme $B_1 - B_2 - R = 0$, on en déduit que le quatrième bit du résultat est 0 et R prend la valeur 0.
9. Le résultat est donc 0010.

2 Protocole

1. Une fois la connexion établie, le serveur commence par envoyer un premier message annonçant le début du défi :

-- Début du défi : Soustraction --

Ce message n'attend pas de réponse.

2. Le serveur envoie ensuite une série de nombres binaires (de taille aléatoire) deux par deux (le premier étant toujours plus grand que le deuxième).
3. Pour chaque paire de nombres binaires, le serveur doit recevoir en retour un nombre binaire (sous forme binaire) égal à la différence des deux nombres binaires envoyés.
4. Après chaque réponse, le serveur enverra un message commençant par "OK" ou "NOK" suivant si la réponse est correcte ou non.

5. A la fin du défi, le serveur enverra un message indiquant "Défi validé" ou "Défi échoué!". Aucune réponse n'est attendue.
6. Le serveur terminera la communication par le message "FIN", votre client devra alors fermer la socket. Aucune réponse n'est attendue.

3 Exemple de communication

Voici un exemple (incomplet) d'une communication pour ce défi. Dans cet exemple les "<" et ">" indiquent le sens de transfert de chaque message et ne doivent pas être présents dans la communication.

```
< -- Début du défi : Soustraction --  
< 1100010100010100100011011  
< 1100010100010100100011000  
> 11  
< OK  
< 1011111000010111  
< 1011110111010011  
> 1000100  
< OK  
< 110101010000111100000100010  
< 110101010000001000111001010  
> 110011001011000  
< OK
```