

Décalage

1 Descriptif

L'objectif de ce défi est d'implémenter une méthode rajoutant n 0 au début du nombre binaire, le décalant donc de n bits "vers la gauche". Cette opération correspond à une multiplication par 2^n mais peut (doit) être réalisée bien plus efficacement que par une multiplication classique.

2 Protocole

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

-- Début du défi : Décalage --

Ce message n'attend pas de réponse.

2. Le serveur envoie ensuite une série de pairs de messages. Chacune de ces pairs sera composée d'un nombre binaire et d'un nombre entier (le décalage).
3. Pour chaque paire, le serveur doit recevoir en retour un nombre binaire (sous forme binaire) en faisant le décalage du nombre binaire par le nombre entier.
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 : Décalage --
< 1110111110100111101010001
< 0
> 1110111110100111101010001
< OK
< 11010100101101001010101
< 3
> 11010100101101001010101000
< OK
< 100000011101010001000
< 5
> 10000001110101000100000000
< OK
```