

1 Descriptif

L'objectif de ce défi est d'implémenter une méthode calculant le pgcd du nombre binaire courant avec celui donné en paramètre. Nous utiliserons bien évidemment l'algorithme d'Euclide pour cela, mais attention, en raison de la taille de nombres manipulés, il n'est pas envisageable d'utiliser la version récursive de cet algorithme.

Pour rappel, l'algorithme d'Euclide fonctionne de la façon suivante. Soient a et b deux nombres entiers positifs.

1. si $a < b$ on échange a et b .
2. Tant que b n'est pas nul, on remplace b par a modulo b et a par b .
3. Quand b est nul, on renvoie a .

2 Protocole

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

-- Début du défi : PGCD --

Ce message n'attend pas de réponse.

2. Le serveur envoie ensuite une série de nombres binaires (de taille aléatoire) deux par deux.
3. Pour chaque paire de nombres binaires, le serveur doit recevoir en retour un nombre binaire (sous forme binaire) égal au PGCD 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 : PGCD --  
< 1000  
< 1010111000010000100010  
> 10  
< OK  
< 10111100  
< 0  
> 10111100  
< OK  
< 1110110100  
< 1000011011110111001  
> 1  
< OK
```