

1 Présentation

Ce fichier contient quelques indications sur les algorithmes de conversion depuis ou vers un MotBinaire ainsi que des indications pour les phases 2 et 3 du module de conversion.

2 Algorithmes de conversion

2.1 D'un Long vers un MotBinaire

La taille du mot binaire est forcément 32. Il suffit alors d'effectuer la décomposition en binaire du long L donné en paramètre : on regarde L modulo 2 pour fixer le bit puis on divise L par 2 pour passer au bit suivant. Il est aussi possible d'utiliser la méthode static `valueOf` de la classe `BitSet`.

2.2 D'un byte vers un MotBinaire

La taille du mot binaire est forcément 8. La classe `BitSet` dispose d'une méthode static `valueOf` acceptant un `byte[]` en entrée.

2.3 D'un caractère vers un MotBinaire

La taille du mot binaire est forcément 8 (on travaille en UTF-8). Il suffit alors de récupérer le code UTF-8 du caractère et de faire comme pour un long.

2.4 D'un MotBinaire vers un String

Il faut prendre le motbinaire par groupe de 8 bits. Pour chacun de ces groupes, on calcule l'entier associé que l'on convertit en un caractère et que l'on place dans une chaîne de caractère.

2.5 D'un MotBinaire vers un entier

On rappelle qu'en binaire le mot abc vaut : $c2^0 + b2^1 + a2^2$.

3 Indications pour les phases 2 et 3

3.1 Descriptif des phases

L'analyse du module de conversion du logiciel nous a permis de déterminer que celui-ci est composé de deux communications distinctes (phase 2 et phase 3).

Lors de la première communication le logiciel génèrera progressivement une suite de longueur aléatoire composée de "longs" et de caractères. Le logiciel semble attendre en retour ces mêmes éléments convertis en binaire. Ainsi, si le logiciel envoie "1231231123", il attendra en retour : "01001001011000110001100010010011".

Lors de la seconde communication le logiciel envoie une séquence de dix mots binaires dont les 5 premiers devront être convertis en entiers et les 5 derniers en chaînes de caractères.

Bien évidemment, chacune des communications devra commencer par l'envoi du mot de passe de celle-ci et se terminera par l'envoi par le logiciel de l'instruction "END".

3.2 Indications pour les mots de passes des communications

Le mot de passe de la phase 2 a été encodé par une variante de l'algorithme de transposition où l'on effectue sur les lignes du tableau la même permutation que sur les colonnes. La clé de transposition est "ANAKIN" et la clé chiffrée est :

YKOTO!U_PRT_S_RTL_AEM_OEOOO_HS_RI_

Le mot de passe de la phase 3 a été encodé par un algorithme simple donnant :

01000100010000001010100100100101101010110010000001010001000100111101010010