===[updated-1]===

Θέμα 9. (15%)

(i) (5%) Σας δίνω μια φράση (χωρίς κενούς χαρακτήρες) η οποία έχει κρυπτογραφηθεί με χρήση ενός LFSR-10 bit (αφού πρώτα χρησιμοποίησα την 5-bit κωδικοποίηση για να το μετατρέψω σε μια ακολουθία από bits). Η feedback function που χρησιμοποίησα είναι,

$$. \ x^{10} + x^9 + x^7 + x^6 + 1$$

Επίσης, δίνεται η κρυπτογράφηση του ab : **Enc(ab)=sq**

Αποκρυπτογραφήστε το κείμενο που
υπάρχει στο zip file με το όνομα lfsr1.txt.

Υποδ. Η κρυπτογράφηση του ab δεν έγινε με τα πρώτα 10-bits που έδωσε το lfsr-10 (δηλ. το seed) αλλά με τα bits από την θέση 10 εώς και 19.
Για να βρείτε το seed πρέπει να λύσετε ένα σύστημα 10X10. Προσοχή το sq→1001010000 που είναι η έξοδος του lfsr δεν είναι το keystream.
Πρεπει να βρειτε το keystream από τις εξισώσεις
bits(msg)+keystream = bits(sq) και κατόπιν την εσωτερική κατάσταση του lfsr. Το σύστημα που θα λύσετε θα είναι της μορφης Bx=y, όπου y η internal state στον 10ο γύρο και x το ζητούμενο seed.

(ii) (15%) Στο zip file του project βρίσκεται επίσης και το αρχείο με όνομα lfsr2.txt
Αυτό έχει προκύψει από τον συνδυασμό των παρακάτω δύο lfsr's.

$$x^{10} + x^9 + x^7 + x^6 + 1,$$

$$x^{16} + x^8 + x^7 + x^3 + x^2 + 1$$

Η κρυπτογράφηση του κειμένου έγινε ως εξής.
1. Το άρχικο κείμενο μετατράπηκε σε bits σύμφωνα με τον Πίνακα 1.
2. Οι έξοδοι των δύο lfsr's (με κάποια seed άγνωστα σε εσάς) έγιναν xor.
3. Θεωρώ ένα keystream (όπως προέκυψε από το βήμα 2) μήκους όσο το μήκος του μηνύματος (σε bit),
4. Το μήνυμα γίνεται xor με το keystream και μετατρέπεται σε λέξη σύμφωνα με τον πίνακα 1.
5. Επίσης, δίνεται η κρυπτογράφηση του abcd : **Enc(abcd)=!c.)**
(η κρυπτογράφηση του γνωστού κειμένου abcd έγινε με το keystream που προέκυψε από τα 2-lfsr, από τις θέσεις 10 εώς και 29)

Βρείτε το αρχικό κείμενο.