Cryptography

Carl Svensson

Hard Task

Because her attempts to use the substitution cipher were not successful, Heidi is revisiting some old saved messages encrypted using a different algorithm. To her despair, she has lost her secret key, which she used to decrypt and encrypt all the messages.

The crypto system works as follows. Every character is treated as a number as listed in the table on the second page. Both the message and the key consists of n characters. Encryption works by adding, modulo 27, to each character in the message the corresponding character in the key. Decryption works the same way by subtracting the key from the ciphertext to recover the original message.

Find a formal definition on the second page of the statement.

You are given some of Heidi's messages, encrypted with Heidi's key, in the file `messages.in`. Your task is to write a function which can decrypt *any* ciphertext of length at most 80 that was encrypted with *that particular key of Heidi*. The program will take as input a number m followed by m ciphertexts, each on its own line. The program should output m lines with the decrypted messages. Note that if a ciphertext is k characters long, only the first k characters of the key have been used to encrypt the message and thus you should return a message of k characters.

Tâche Difficile

Puisque t'as réussi à convaincre Heidi que son nouvel algorithme n'est effectivement pas si bon qu'elle ne l'espérait, elle parcourt quelques vieux messages cryptés par un autre algorithme. Malheureusement elle ne se rappelle plus de la clé d'encryptage qu'elle avait utilisée pour encrypter et décrypter ses messages.

Ce qui est sûr est que le système de cryptage fonctionne de la manière suivante. Chaque caractère est traité comme un nombre, comme l'indique la table sur la deuxième page. Le message et la clé contiennent n caractères. Pour encrypter, on ajoute à chaque caractère du message le caractère de la clé à la même position, et le résultat modulo 27 donne le message crypté. T'aurais deviné que ce message encrypté peut être décrypté en soustrayant la clé (modulo 27).

Tu peux trouver une definition formelle sur la deuxième page.

On te donne le fichier `messages.in`, qui contient quelques-uns des messages de Heidi, encryptés avec sa clé. Tu dois maintenant écrire une fonction qui peut décrypter *n'importe quel* texte encrypté, de longueur 80 caractères maximal, qui a été encrypté avec *cette même clé*. Le programme prendra en entrée le nombre m, suivi de m textes encryptés, un par ligne. Le programme doit produire en sortie m lignes avec les messages décryptés. Nota Bene, si le message encrypté est de longeur k, seulment les k premières caractères de la clé ont été utilisé pour encrypter le message, et donc tu dois répondre avec une message de longeur k.

Schwierige Aufgabe

Nachdem du Heidi überzeugen konntest dass ihr Verschlüsslungssystem doch nicht so sicher ist wie erhofft, widmet sie sich einigen älteren verschlüsselten Botschaften, die auf einem anderen Algorithmus beruhen. Leider kann sie sich nicht mehr an den Schlüssel zum ver- und entschlüsseln erinnern.

Das Verschlüsslungssystem funktioniert folgendermassen: Jedes Zeichen wird als Zahl betrachtet, wie in der untenstehenden Tabelle aufgeführt. Sowohl die Botschaft, wie auch der Schlüssel bestehen aus n Zeichen. Ein Klartext wird verschlüsselt, indem zu jedem Zeichen das entsprechende Zeichen des Schlüssels addiert wird. Dieses Resultat, modulo 27, ergibt den Geheimtext. Dieser Geheimtext kann dann wieder entschlüsselt werden, indem der Schlüssel abgezogen wird (modulo 27).

Für eine formelle Definition, siehe unten.

Wir geben dir einige von Heidis Geheimbotschaften, die mit ihrem Schlüssel verschlüsselt worden sind. Du findest sie in `messages.in`. Deine Aufgabe besteht darin eine Funktion zu schreiben, welches *jeden beliebigen* Geheimtext von bis zu 80 Zeichen entschlüsseln kann, wenn dieser durch *exakt diesen Schlüssel von Heidi* entstanden ist. Dein Programm erhält als Eingang die Zahl m, gefolgt von m Zeilen. Jede Zeile enthält einen Geheimtext, für welchen dein Programm den Klartext ausgeben muss, ein Klartext pro Zeile. Nota Bene, falls der Geheimtext k Zeichen lang ist, wurden nur die ersten k Buchstaben des Schlüssels zum Verschlüsseln verwendet, entsprechend sollte die Länge des Klartextes ebenfalls k sein.

Character values / Valeurs de caractères / Zeichenwerte

Character(s)	Value(s)
`_`	`0`
`a-z`	`1-26`

Formal definition / Définition formelle / Formelle definition

$$
\begin{aligned}
k &\in \{\_, a, b, \ldots, z\}^n \\
m &\in \{\_, a, b, \ldots, z\}^n \\
c_i &= \mathrm{enc}_k(m_i) = m_i + k_i \mod 27 \\
m_i &= \mathrm{dec}_k(c_i) = c_i - k_i \mod 27
\end{aligned}
$$

Sample / Exemple / Beispiel

Note that, in order to not reveal Heidi's secret, this sample run uses a different key, not the one that Heidi used. The key in this case is `brownie`.

Afin de ne pas révéler la clé secrète de Heidi, cet exemple utilise une clé différente à celle utilisée par Heidi. La clé en question est `brownie`.

Damit wir hier nicht schon alles verraten, gebraucht dieses Beispiel einen anderen Schlüssel als Heidis Geheimschlüssel. In diesem Fall ist `brownie` der Schlüssel.

Input / Entrée / Eingabe	Output / Sortie / Ausgabe
`3`	`message`
`owgoopj`	`secret`
`uwrnsb`	`a_cow`
`cqrkj`	