P2002 對稱式加密演算法

網際網路發達後，訊息在網路上流通很容易被竊取，資料加密後再傳送實有必要；加密後傳送的資料必須確保唯有當資料抵達正確的接收方後，才能夠經解密過程還原原文，如此才能真正達到機密不外洩的目的。

以下為對稱式加密的基本概念，加密者與解密者持有共同的密鑰（Secrete key），加密者以該密鑰加密，解密者亦以同樣的密鑰解密，甚至加解密程式也完全一樣，將介紹的 XOR 加解密過程即具備如此特性；例如，茲利用一十六進位密鑰 AC（10101100）對一 byte 的明文資料（01110110）加密，如下：

$$\begin{array}{rr} & 01110110\text{（明文）} \\ XOR & 10101100\text{（密鑰）} \\ \hline & 11011010\text{（密文）} \end{array}$$

經加密後，密文與明文顯然不同，接收端只要對密文用相同的密鑰進行 XOR 運算，便可還原，如下：

$$\begin{array}{rr} & 11011010\text{（密文）} \\ XOR & 10101100\text{（密鑰）} \\ \hline & 01110110\text{（明文）} \end{array}$$

XOR 加解密演算法雖然簡單，但若密鑰只有一個 byte，要破解並不困難，例如以暴力法或字元頻率分析，不難解密；為增加解密的困難度，本題中所給的密鑰將有 n（$1 \le n \le 128$）個 bytes（$k_1 k_2 \cdots k_n$）；加解密時，對明文或密文中的內容從 k_1 開始，依序用不同密鑰中的 byte 施以 XOR 運算，密鑰用到 k_n 後，又回到 k_1，直到明文或密文結束。

本題中，你將被要求對一個檔案的內容進行加密或解密，程式中你須要正確使用不同檔案應被開啟的模式；若要求的工作是解密，你除了將解密後的明文寫入指定的檔案中外，也須將其內容輸出至 stdout；若所要求的工作是加密，你除了將加密後的密文寫入指定的檔案中外，也須將其內容以十六進制方式輸出至 stdout，輸出方式見稍後說明與範例輸出。

輸入說明

測資第一行依序為一字元（'d'或'e'）與兩個檔名（file1 與 file2），字元'd'表示進行解密，字元'e'表示加密，file1 為讀入的明文或密文的檔名，file2 為寫入的明文或密文的檔名；第二行為密鑰資訊，該行最前方為一整數說明密鑰的長度 n，隨後為 n 個以空白間隔之密鑰值 k_1，k_2，…，k_n，密鑰值均為 0~255 的十進位數字。

輸出說明

依測資要求讀入明文或密文檔案 file1，加解密後將明文或密文寫入檔案

file2；若為解密工作，file2 內容亦須輸出至 stdout。若為加密工作，需將密為內容以二進制方式輸出至 stdout，每行輸出 16 bytes（最後一行除外），並以空白區隔每個 byte。

以下範例假設明文檔名為 plain.txt，檔案內容為；

"CodingPass,\nNice to meet you!\n"

假設密文檔名 cypher.bin 之內容為以上檔案加密後之二進制檔案。

以下範例輸出中將僅呈現 stdout 的結果。

範例輸入(I)

```
e plain.txt cypher.bin
2 121 212
```

範例輸出(I)

```
3A BB 1D BD 17 B3 29 B5 0A A7 55 DE 37 BD 1A B1
59 A0 16 F4 14 B1 1C A0 59 AD 16 A1 58 DE
```

範例輸入(II)

```
d cypher.bin any.txt
2 121 212
```

範例輸出(II)

```
CodingPass,
Nice to meet you!
```