

Sequências de DNA

Por Cláudio L. Lucchesi  Brasil

Timelimit: 4

Thomas, um cientista da computação que trabalha com sequências de DNA, precisa computar as maiores subsequências comuns de dados pares de strings. Considere um alfabeto S de letras e uma palavra $w = a_1a_2 \dots a_r$, onde $a_i \in \Sigma$, para $i = 1, 2, \dots, r$. Uma subsequência de w é uma palavra $x = a_{i_1}a_{i_2} \dots a_{i_s}$ tal que $1 \leq i_1 < i_2 < \dots < i_s \leq r$. A subsequência x é um segmento de w se $i_{j+1} = i_j + 1$, para $j = 1, 2, \dots, s-1$. Por exemplo a palavra *ove* é um segmento da palavra *lovely*, enquanto a palavra *loly* é uma subsequência de *lovely*, mas não um segmento.

Uma palavra é uma subsequência comum de duas palavras w_1 e w_2 se ela é uma subsequência de cada uma das duas. Uma maior subsequência comum de w_1 e w_2 uma subsequência comum de w_1 e w_2 tendo o maior comprimento possível. Por exemplo, considere as palavras $w_1 = \text{lovxxelyxxxxx}$ e $w_2 = \text{xxxxxxxlovely}$. As palavras $w_3 = \text{lovely}$ e $w_4 = \text{xxxxxxx}$, a última de comprimento 7, são ambas subsequências comuns de w_1 e w_2 . De fato, w_4 é a maior subsequência comum delas. Perceba que a palavra vazia, de comprimento zero, é sempre uma subsequência comum, apesar não ser necessariamente a mais longa.

No caso do Thomas, existe um requerimento extra: a subsequência tem que ser formada de segmentos comuns tendo comprimento K ou maior. Por exemplo, se Thomas decidir que $K = 3$, então ele considera *lovely* como uma subsequência comum aceitável de *lovxxelyxxxxx* e *xxxxxxxlovely*, enquanto *xxxxxxx*, que tem um comprimento de 7 e também é uma subsequência comum, não é aceitável. Você pode ajudar Thomas?

Entrada

A entrada consiste de vários casos de teste. A primeira linha de um caso de teste contém um inteiro K representando o comprimento mínimo de segmentos comuns, onde $1 \leq K \leq 100$. As próximas duas linhas contém, em cada, uma palavra com letras minúsculas do alfabeto tradicional de 26 letras. O comprimento L de cada palavra satisfaz a desigualdade $1 \leq L \leq 10^3$. Não existem espaços nas linhas de entrada. O final da entrada é indicado por uma linha contendo um zero.

Saída

Para cada caso de teste na entrada, seu programa deve imprimir uma única linha, contendo o comprimento da maior subsequência formada por segmentos consecutivos de comprimento de pelo menos K de ambas palavras. Se não existir uma subsequência comum de comprimento maior que zero, então deve ser imprimido 0.

Exemplo de Entrada	Exemplo de Saída
3 lovxxelyxxxxx xxxxxxxlovely 1 lovxxelyxxxxx xxxxxxxlovely 3 lovxxxelxyxxxx xxxlovelyxxxxxxx 4 lovxxxelyxxx xxxxxxxlovely 0	6 7 10 0

