

### Problem 3—Approximate Matches

Given a non-null string of text  $S$ , a non-null string of text  $P$ , and an integer  $k \geq 0$ , find all non-null substrings of  $S$  that match or “approximately match”  $P$  with at most  $k$  mismatches. A substring is considered to *match* the string  $P$  if they are identical (as you might expect), and *approximately match* if the strings differ by no more than  $k$  mismatches, as defined next. If a minimum of  $m$  characters have to be removed from  $T$  to make it identical to  $P$ , then  $T$  approximately matches  $P$  with  $m$  mismatches. If a minimum of  $n$  characters have to be removed from  $P$  to make it identical to  $T$ , then  $T$  approximately matches  $P$  with  $n$  mismatches. If  $m$  characters have to be removed from  $T$  and  $n$  characters have to be removed from  $P$  to make  $T$  and  $P$  identical, then  $T$  approximately matches  $P$  with  $m+n$  mismatches (here  $m+n$  should be minimized).

Consider the string  $S = \text{“The sky is blue”}$ .

The substring “The sky” in  $S$  matches the string “The sky”.

The substring “The sky” in  $S$  approximately matches the string “The skye” with one mismatch (the extra “e” in the string).

The substring “sky is” in  $S$  approximately matches the string “skis” with two mismatches (the characters “y” and “ ” in the substring).

The substring “sky is” in  $S$  approximately matches the string “tennis” with eight mismatches (four characters are removed from “sky is” and four are removed from “tennis”).

As another example, the string “scrapple” approximately matches the string “apples” with four mismatches (by removing s, c, r from “scrapple” and s from “apples”). Although there are other ways to remove letters from these two strings to make the results identical (for example removing all but the s from both strings), four characters is the minimum number that can be removed.

Note that case is important in this problem; thus T does not match t, but T approximately matches t with two mismatches.

**INPUT SPECIFICATION.** Each data set consists of three lines that contain  $k$ ,  $S$ , and  $P$ , respectively.  $k$  is an integer;  $S$  is a string of length between 1 and 50 inclusive, and  $P$  is a string of length between 1 and 20 inclusive. All lines will be terminated by  $\langle \text{EOLN} \rangle$ . The line containing  $k$  will contain no spaces. The lines containing  $S$  and  $P$  may well contain spaces, but all spaces are considered part of the respective string. Your program should end when it encounters a negative  $k$ , in this case,  $S$  and  $P$  need not even be present.

**OUTPUT SPECIFICATION.** For each match with  $i$  mismatches ( $0 \leq i \leq k$ ), your program must generate a line of one of the following forms (whichever is appropriate):

$Q$  matches  $P$

$Q$  matches  $P$  with 1 mismatch

$Q$  matches  $P$  with  $i$  mismatches

where  $Q$  is a substring of  $S$  that matches  $P$  or that approximately matches  $P$  with  $i$  mismatches. If more than one substring in  $S$  matches or approximately matches  $P$ , they must be printed in the order they appear in  $S$  (from left to right). Substrings that start at the same position must be listed

in the order of their size (shortest to longest). No value for  $Q$  may be listed more than once. If there is no match, no output should be generated. There is an extra **<EOLN>** after the output for each data set (even if there is no output for the data set).

### **SAMPLE INPUT.**

```
1<EOLN>
He·did·not·care·about·the·carpet·in·the·car.<EOLN>
car<EOLN>
2<EOLN>
ABC<EOLN>
BC<EOLN>
-1<EOLN>
<EOF>
```

### **SAMPLE OUTPUT.**

```
·car·matches·car·with·1·mismatch<EOLN>
ca·matches·car·with·1·mismatch<EOLN>
car·matches·car<EOLN>
care·matches·car·with·1·mismatch<EOLN>
ar·matches·car·with·1·mismatch<EOLN>
carp·matches·car·with·1·mismatch<EOLN>
car.·matches·car·with·1·mismatch<EOLN>
<EOLN>
AB·matches·BC·with·2·mismatches<EOLN>
ABC·matches·BC·with·1·mismatch<EOLN>
B·matches·BC·with·1·mismatch<EOLN>
BC·matches·BC<EOLN>
C·matches·BC·with·1·mismatch<EOLN>
<EOLN>
<EOF>
```