

## Problem 1—Internal Palindromes

Elven is a very complex language because it relies not only on subtext but also on substrings. Different patterns of letters within a piece of text can add to the meaning of the overall text. In particular, the elves attach a lot of symbolism to palindromes. This problem requires you to locate palindromes within a line of text. Given a line of text, you are to find the *longest* palindrome that is a subsequence of the line. That is, you are to eliminate certain (not necessarily adjacent) letters from the text so that the letters remaining, in their original relative order, form the longest possible palindrome.

**INPUT SPECIFICATION.** The input file will consist of a number of strings of one or more capital letters. Each string will be followed by <EOLN>. An extra <EOLN> will follow the last string in the file.

**OUTPUT SPECIFICATION.** The output cases should appear in the same order as the corresponding input cases. Each output case should be in the form “Case *c*:” (where *c* is the case number), followed by two <EOLN>'s, followed by the longest palindrome that is a subsequence of the input case. If there are more than one palindrome of maximum length, all valid answers must be printed exactly once, in alphabetical order, each followed by <EOLN>. An extra <EOLN> should follow each output case.

### **SAMPLE INPUT.**

```
RADARS<EOLN>
BATMAN<EOLN>
<EOLN>
<EOF>
```

### **SAMPLE OUTPUT.**

```
Case 1:<EOLN>
<EOLN>
RADAR<EOLN>
<EOLN>
Case 2:<EOLN>
<EOLN>
AMA<EOLN>
ATA<EOLN>
<EOLN>
<EOF>
```