Problem 2—Palindromic Sums

Otto Preminger's name (Otto) is cool because it's a palindrome, meaning it reads the same backwards and forwards. Numbers can be palindromes, too. At one time, people thought that if you added a number to itself backwards, and then added the sum to itself backwards, and so forth, you'd eventually get to a numeric palindrome. For example: 987+789=1776. 1776+6771=8547. 8547+7458=16005. 16005+50061=66066! And, who knows, maybe you will eventually hit a palindrome guaranteed, but so far, no one has managed to get a palindrome out of 196, and most modern mathematicians suspect the hypothesis to be false.

Given a positive number, you are to print the palindrome that eventually results from repeatedly adding a number to its reverse. (If the starting number is a palindrome, you still have to do at least one sum.) And numbers, such as 196, with no known palindromic sum will not be given. All inputs will yield an answer (eventually).

INPUT SPECIFICATION. You will be given a set of input cases ,each of which will be an unsigned decimal integer less than 100000. The last input case will be followed by -1. There may be any number of spaces and **<EOLN>**'s before, after, and between the input cases and the terminating -1.

OUTPUT SPECIFICATION. The output cases should appear in the same order as the input cases. Each output case will be of the form “Case *c*: The palindromic sum is *s*” (where *c* is the number of the input case and *s* is the answer) followed by **<EOLN>**.

SAMPLE INPUT.

```
27<EOLN>
987<EOLN>
-1<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case•1:•The•palindromic•sum•is•99<EOLN>
Case•2:•The•palindromic•sum•is•66066<EOLN>
<EOF>
```