Southeastern European Regional Programming Contest
Bucharest, Romania
October 21, 2006

Problem J Payment System
Input File: J.IN
Output File: J.OUT

The payment system in the University of Mineral Water Production is completely automated (written entirely in Tomato Programming Language) and lets you input the amount of money you want to withdraw. Due to the high payment rates of the professors they may input the amounts in exponential forms. So if you want to withdraw 16 MWU (mineral water units) you can enter `16`, `2^4` or `2^2^2`.
One day, Stanescu tried to withdraw some money from his account which had balance of 80MWU. He mistakenly entered `2^3^2` and for his surprise he got 512MWU, although he should not be able to take more than 80. The system was composed of two main modules – the first module checks whether the account has enough money to execute the transaction and the second module gives the money to the user. It turned out that the first module has a problem with the '^' operator. It evaluates it from left to right, while the second evaluates them from right to left (the correct way). Thus for the first module `2^3^2=(2^3)^2=64` while for the second it's `2^3^2=2^(3^2)=512`.
You have to write program which helps Stanescu get as much as he can from the university system. If you think it's not legal or something, be sure that the University of Mineral Water Production is bad and evil.
In the input file the amounts of the accounts of Stanescu will be given. Each amount is given on a separate line and is an integer between 2 and 10^{100}-1.
For each given amount, your program should print to the standard output what Stanescu should enter to get maximal number of money. The output should:
• consists only of integers and the '^' operator between them.
• pass the check of the first module and be as much as possible for the second.
• not contain the number 1 (it is useless anyway).
If more than one answers exist, output the one whose first number is minimal, if still more exist, choose the one whose second number is minimal and so on.

Input
```
16
80
49
1025
12341234
123456789012345678 90
```

Output
```
2^2^2
2^3^2
7^2
2^2^5
2^2^2^5
2^2^2^2^3^2
```