Problem G: A to Z Numerals

Source file: `numeral.{c, cpp, java}`
Input file: `numeral.in`

Roman numerals use symbols `I`, `V`, `X`, `L`, `C`, `D`, and `M` with values 1, 5, 10, 50, 100, 500, and 1000 respectively. There is an easy evaluation rule for them:

Rule Δ: Add together the values for each symbol that is either the rightmost or has a symbol of no greater value directly to its right. Subtract the values of all the other symbols.

For example: `MMCDLXIX` = 1000 + 1000 - 100 + 500 + 50 + 10 - 1 + 10 = 2469.

Further rules are needed to uniquely specify a Roman numeral corresponding to a positive integer less than 4000:

1. The numeral has as few characters as possible. (`IV` not `IIII`)
2. All the symbols that make positive contributions form a non-increasing subsequence. (`XIV`, not `VIX`)
3. All subtracted symbols appear as far to the right as possible. (`MMCDLXIX` not `MCMDLIXX`)
4. Subtracted symbols are always for a power of 10, and always appear directly to the left of a symbol 5 or 10 times as large that is added. No subtracted symbol can appear more than once in a numeral.

Rule 4 can be removed to allow shorter numerals, and still use the same evaluation rule: `IM` = -1 + 1000 = 999, `ILIL` = -1 + 100 + -1 + 100 = 198, `IVL` = -1 -5 + 100 = 94. This would not make the numerals unique, however. Two choices for 297 would be `CCVCII` and `ICICIC`. To eliminate the second choice in this example, Rule 4 can be replaced by

> 4'. With a choice of numeral representations of the same length, use one with the fewest subtracted symbols.

Finally, replace the Roman numeral symbols to make a system that is more regular and allows larger numbers: Assign the English letter symbols `a`, `A`, `b`, `B`, `c`, `C`, …, `y`, `Y`, `z`, and `Z` to values 1, 5, 10, 5(10), 10^2, (5)10^2, …, 10^{24}, (5)10^{24}, 10^{25}, and (5)10^{25} respectively. Though using the *whole* alphabet makes logical sense, your problem will use only symbols `a-R` for easier machine calculations. (`R`= (5)10^{17}.)

With the new symbols `a-Z`, the original formation rules 1-3, the alternate rule 4', and the evaluation rule Δ, numerals can be created, called A to Z numerals. Examples: `ad` = -1 + 1000 = 999; `aAc` = -1 - 5 + 100 = 94.

Input: The input starts with a sequence of one or more positive integers less than (7)10^{17}, one per line. The end of the input is indicated by a line containing only 0.

Output: For each positive integer in the input, output a line containing only an A to Z numeral representing the integer.

Do not choose a solution method whose time is exponential in the number of digits!

Example input:	Example output:
999 198 98 297 94 666666666666666666 0	ad acac Acaaa ccAcaa aAc RrQqPpOoNnMmLlKkJjIiHhGgFfEeDdCcBbAa

Last modified on October 18, 2009 at 9:40 AM.