Ancient Pocket Calculator

Adam likes pocket calculators, especially the early ones. As one of his favorite calculators is about 40 years old, he is not sure how long he will be able to use it. So he had the ingenious idea to develop a simulator that behaves exactly like his calculator. This simulator must be able to read a sequence of keystrokes from the calculator's keypad, process the appropriate calculations and print the calculator's output. As Adam needs only basic arithmetic, the following keys will be sufficient: digits 0 to 9, decimal point, operators +, -, x and : (for division), the equal sign for calculating and displaying a result and the [C] key for reseting the calculator and clearing the display, i.e. the display is set to "0.". Calculations are done from left to right without any operator precedence.

You may call Adam's calculator a headstrong comtemporary, because of its special behaviour: There is no invalid sequence of keystrokes. You can press arbitrary keys one after another, the calculator always knows how to handle it. If more than one operator key (including [=]) is pressed directly after another, only the last of these operators will be processed - all previous ones (in that continuous sequence) are ignored.
If more than 8 digit keys are pressed for the input of a single number, only the first 8 digits will be processed - all following digits are ignored. If the actual display value is zero, the typing of the zero key will have no effect, it's just ignored like successive keystrokes of the same operator. If a floating point value is typed in, a leading zero directly before the decimal point may be left out, but will be displayed just the same. If the decimal point key is pressed within a number that already has a decimal point typed in or if the input of a number (as a sequence of digit keys) is terminated by a decimal point, that has no effect.

Input

Input starts with a positive integer t (t<1000) in a single line, the number of testcases. Then t lines follow, each line giving the description of an arbitrary sequence of keystrokes on the calculator's keypad. Every key is enclosed by square brackets, all keystrokes are separated by a single space. The number of keystrokes per sequence is less than 500 and every sequence will be terminated by [=].

Output

For each sequence of keystrokes print the result the calculator will show on the display after the complete sequence of keystrokes has been processed. The size of the display is 8 digits plus an optional "-"-sign in front of the leftmost digit and a decimal point that will always appear, even if the result is an integer value. If a value has more than 8 decimal digits, it has to be rounded to fit into 8 digits. As the calculator's display is filled from right to left, the output has to be adjusted to the right.

If the absolute value of a number rounded to an integer needs more than 8 digits, scientific notation is used. Same case, if the absolute value of a number is larger than 10^{-100}, but rounding to 8 digits would result in displaying zeros only. If the absolute value is not larger than 10^{-100}, it results to zero.
In scientific notation a number is expressed as a product of a decimal part and a power of 10. The decimal part has always exactly one non-zero digit before the decimal point, an optional "-" sign in front of the leftmost digit and upto 4 digits after the decimal point, rounded if necessary. If the exponent is negative, a "-" follows, otherwise a space. Then follow two digits representing the exponent; a leading zero is shown in the exponent, if necessary.

Notice that there are two cases, where the calculator will display "Error." instead of showing a result. If a (final or interim) result has a rounded absolute value of at least 10^{100} or if you divide by zero. After an error has occured, all following keystrokes are ignored unless [C] is pressed.

For more clarity of the calculator's behaviour and the required input and output format please look at the examples below.

Example

Input:
12
[3] [+] [4] [x] [5] [=]
[1] [:] [6] [=]
[4] [.] [8] [-] [x] [+] [-] [+] [x] [-] [.] [=]
[4] [.] [8] [-] [x] [+] [-] [+] [x] [.] [=]
[+] [+] [+] [+] [+] [+] [1] [=] [=] [=]
[9] [8] [C] [-] [7] [6] [5] [4] [3] [2] [1] [0] [1] [2] [3] [4] [=]
[1] [=] [2] [=] [3] [=]
[5] [:] [9] [8] [7] [8] [9] [8] [9] [8] [7] [8] [8] [:] [4] [5] [6] [7] [8] [9] [=]
[-] [9] [9] [9] [9] [9] [9] [9] [9] [-] [-] [-] [-] [8] [8] [8] [8] [8] [8] [=]
[2] [.] [3] [.] [4] [.] [5] [=]
[0] [0] [0] [0] [0] [=]
[.] [:] [.] [=]

Output:
```
      35.
 0.1666667
      4.8
       0.
       1.
-76543210.
       3.
  1.108-13
-1.0089 08
     2.345
       0.
    Error.
```