Spreadsheet

`sheet.p`, `sheet.c`, `sheet.C`

In 1979, Dan Bricklin and Bob Frankston wrote VisiCalc, the first spreadsheet application. It became a huge success and, at that time, was the killer application for the Apple II computers. Today, spreadsheets are found on most desktop computers. And, with the admission to the European Union, their importance even raises. There are important computations to be performed by businessmen. For instance, they may want to determine, whether it is better to buy a Czech product or some of its variants made in other countries.

The idea behind spreadsheets is very simple, though powerful. A spreadsheet consists of a table where each cell contains either a number or a formula. A formula can compute an expression that depends on the values of other cells. Text and graphics can be added for presentation purposes.

You are to write a very simple spreadsheet application. Your program should accept several spreadsheets. Each cell of the spreadsheet contains either a numeric value (integers only) or a formula, which only support sums. After having computed the values of all formulas, your program should output the resulting spreadsheet where all formulas have been replaced by their value.

A1	B1	C1	D1	E1	F1	...
A2	B2	C2	D2	E2	F2	...
A3	B3	C3	D3	E3	F3	...
A4	B4	C4	D4	E4	F4	...
A5	B5	C5	D5	E5	F5	...
A6	B6	C6	D6	E6	F6	...
...	...	...	...	...	...	...

Input Specification

The first line of the input file contains the number of spreadsheets to follow. A spreadsheet starts with a line consisting of two integer numbers, separated by a space, giving the number of columns and rows. None of these numbers will not exceed 1000. The following lines of the spreadsheet each contain a row. A row consists of the cells of that row, separated by a single space.

A cell consists either of a numeric integer value or of a formula. A formula starts with an equal sign ("`=`"). After that, one or more cell names follow, separated by plus signs ("`+`"). The value of such a formula is the sum of all values found in the referenced cells. These cells may again contain a formula. There are no spaces within a formula.

You may safely assume that there are no cyclic dependencies between cells. So each spreadsheet can be fully computed.

The name of a cell consists of one to three letters for the column followed by a number between 1 and 999 (including) for the row. The letters for the column form the following series: A, B, C, ..., Z, AA, AB, AC, ..., AZ, BA, ..., BZ, CA, ... ZZ, AAA, AAB, AAC, ... AAZ, ABA, ..., ABZ, ACA, ..., ALK. These letters correspond to the numbers from 1 to 999. The top left cell has the name A1. See the figure above.

Output Specification

The output of your program should have the same format as the input, except that the number of spreadsheets and the number of columns and rows are not repeated. Furthermore, you should print one blank after each spreadsheet to terminate it. And, of course, all formulas must be replaced by their values.

Sample Input

```
1
4 3
10 34 37 =A1+B1+C1
40 17 34 =A2+B2+C2
=A1+A2 =B1+B2 =C1+C2 =D1+D2
```

Output for Sample Input

```
10 34 37 81
40 17 34 91
50 51 71 172
```

P.S.: Note that we do not require asymptotically optimal solution to this problem. All large test cases are randomly generated and they do not contain any specially designed patterns.