[A] The Earth is Flat!

Program:	`flat.(c|cpp|java)`
Input:	`flat.in`
Balloon Color:	`Gold`

Description

Consider the following language:

$$expression = \begin{cases} c & \textit{where c is a single, lower-case letter} \\ (\ e_1\ e_2\ \cdots\ e_t\ n\ ) & \textit{zero or more expressions followed by a natural number} \end{cases}$$

The left column of the table below includes sample expressions of this language. Now the flattening of an expression is defined as follows: A single letter is flattened to itself. An expression of the form ($e_1\ e_2\ \cdots\ e_t\ n$) is flattened by concatenating n copies of the concatenation of the flattening of e_i. In other words, if $f(e)$ is the flattening of e, and + symbolizes concatenation, then $f(\ \ (e_1\ e_2\ \cdots\ e_t\ n)\ \ )$ is:

$$\underbrace{f(e_1) + f(e_2) + \cdots + f(e_t)}_{\text{once}} \;+\; \overbrace{f(e_1) + f(e_2) + \cdots + f(e_t)}^{\text{twice}} \;+\; \cdots \;+\; \underbrace{f(e_1) + f(e_2) + \cdots + f(e_t)}_{\text{the n}^{th}\text{ time}}$$

The following tables show some sample expressions and the result of flattening each.

expression	**flattened expression**
`w`	w
`(c 4)`	cccc
`(a (b c 2) 3)`	abcbcabcbcabcbc

Write a program to flatten a given expression.

Input Format

Your program will be tested on one or more test cases. Each test case is made of one correctly formed expression written on a separate line. A `'$'` character identifies the end of line. The last line of the input, which is not part of the test cases, contains a `'$'` by itself (possibly with leading and/or trailing white spaces). Every expression in the input is grammatically correct according to the grammar specified above. Note that an expression may contain leading, trailing, and/or embedded spaces. Such spaces should be ignored. Letters and numbers are separated from each other by at least one space character.

Output Format

For each test case expression, write its flattening on a separate line. There should be no spaces (other than newlines) in the output.

Sample Input/Output

flat.in

```
w$
(c 4)$
(a (b c 2) 3)$
$
```

OUTPUT

```
w
cccc
abcbcabcbcabcbc
```

Problem A: Take Your Vitamins

Manufacturers of food products are required to place nutrition information labels on their packages. A major part of this information is a listing of important vitamins and minerals, listing both the amount of the chemical present in one serving and the percentage of an adult's minimum daily requirement for that chemical.

Write a program to help prepare these nutritional labels by computing that percentage from the information on the amount present in one serving and the amount constituting the minimum daily requirement.

Input

Input consists of one or more lines, each of the form:

```
  A U R V
```

where A is the amount of a vitamin/mineral present in one serving of the food product, U is the units in which A is measured, R is the minimum daily requirement for that vitamin/mineral, measured in the same units as A, and V is the name of that vitamin/mineral.

A and R will be floating point numbers. U will be a string of alphabetic characters with no embedded spaces. V will be a string of characters, possibly including spaces. A, U, R, and V will be separated from one another by exactly one space, and V is terminated by the end of the input line.

End of the input is signaled by a line in which A is negative.

Output

For each line of input data, your program should determine the percentage of the recommended daily requirement being provided for that vitamin/mineral. If it is at least 1%, your program should print a line of the form

```
  V A U P%
```

where V, A, and U are the quantities from the input, and P is the percentage of the minimum daily requirement represented by the amount A.

V should be printed left-justified on the line. A should be printed with 1 digit precision, and P with zero digits precision. V, A, U, and P should be separated by one space each.

After the last such line, your program should print a line stating

```
Provides no significant amount of:
```

followed by a list of the names of all vitamins/minerals which are provided as less than 1% of the minimum daily requirement. These should be printed one name per line, in the order they occurred within the input.

Example

Input:

```
3500.0 iu 5000.0 Vitamin A
60.0 mg 60.0 Vitamin C
0.15 g 25.0 Fiber
109. mg 990. Phosphorus
0.0 mg 1000.0 Calcium
25.0 mg 20.0 Niacin
-1.0 x 0.0 x
```

Output:

```
Vitamin A 3500.0 iu 70%
Vitamin C 60.0 mg 100%
Phosphorus 109.0 mg 11%
Niacin 25.0 mg 125%
Provides no significant amount of:
Fiber
Calcium
```