

Generating Alien DNA

[The original version of this problem (in Spanish) can be found at <http://www.dc.uba.ar/events/icpc/download/problems/taip2012-problems.pdf>]

GigaFarma is one of the largest pharmaceutical companies in the world, and it is currently conducting experiments using alien DNA. Its goal is to produce a chain of alien DNA that will report the maximum possible benefit when commercialized.

A chain of *alien* DNA can be understood as a non-empty sequence of genes connected by links, and in turn each *gene* is a non-empty sequence of bases. Because not every possible sequence of bases corresponds to a valid gene, GigaFarma has created a catalogue of genes that appear in alien DNA, which are the only ones considered valid sequences of bases. Each of these genes has a *value* according to its functionality, and a given chain of alien DNA has a *market value* that is the sum of the values of the genes that compose it.

We will represent the different *bases* with lowercase letters, 'a'-'z', and *links* using a hyphen '-'. In the following example we can see on the left a possible list of genes and their corresponding values; on the right there are some chains of alien DNA that can be formed with these genes, along with their corresponding market values.

hola	(5)	va-como-va	(9 = 2 + 5 + 2)
como	(5)	les	(3)
les	(3)	como-les	(8 = 5 + 3)
va	(2)	como-como-les	(13 = 5 + 5 + 3)
		como-como-como-les	(18 = 5 + 5 + 5 + 3)

GigaFarma can only produce very specific DNA chains, which we call *producible*. These chains are a non-empty sequence of DNA portions that the company can synthesize, joined without any additional links between them. Each *portion* is a sequence of bases and links containing at least one link, but without any consecutive, initial or final links. Each portion has a given *cost*, determined by the difficulty associated to its production, so each producible chain of DNA has a *production cost* that is the sum of the costs of each of the portions that form it. In the following example, we can see on the left a list of DNA portions and their costs; on the right we have some producible chains of DNA that can be formed with these portions, along with their associated production cost.

como-co	(3)	como-co	(3)
mo-co	(8)	mo-les	(4)
mo-les	(4)	hasta-esta-no-sirven	(300 = 200 + 100)
como-como-les	(12)	como-como-les	(7 = 3 + 4, o 12)
ta-no-sirven	(100)	como-como-como-les	(15 = 3 + 8 + 4)
hasta-es	(200)		

Note that there might be multiple ways of forming the same producible chain using different portions. This is the case of "**como-como-les**" in the example, which can be obtained using portions "**como-co**" and "**mo-les**" with a production cost of 7, or just using "**como-como-**

les" with a production cost of **12**. Of course, when there is more than one way to synthesize a given producible chain of DNA, GigaFarma will always do so using the cheapest possible process.

Clearly, the set of alien DNA chains is infinite, just like the set of producible DNA chains. However, GigaFarma is not directly interested in any of these sets, but in their intersection. If we check the previous examples, we can see that "**como-les**" is a valid alien DNA chain, but is not producible; "**mo-les**" is producible, but is not an alien DNA chain; and "**como-como-les**" is both. For each alien and producible DNA chain, the company can commercialize this chain to get a *net benefit* that equals the market value of this chain minus its production cost. Of course, if this net benefit is not positive, the corresponding chain will never be produced.

Because there is so much genetic material all over the place, GigaFarma would pay anything in order to know what is the maximum net benefit it can obtain for some producible and alien DNA chain.

Input

Each test case is described using several lines. The first line contains two integer numbers **G** and **P**, representing the number of genes in the catalogue and the number of portions GigaFarma can produce ($1 \leq G, P \leq 100$).

Each of the following **G** lines describes a different gene using a string **S** and an integer **V**. The string **S** has between **1** and **10** characters, and is formed solely by lowercase letters representing the bases that form this gene; the integer **V** represents the value of this gene ($1 \leq V \leq 1000$).

Each of the following **P** lines describe a different DNA portion, using a string **T** and an integer **C**. The string **T** has between **1** and **30** characters, and is composed of lowercase letters and hyphens only, respectively representing the bases and links in this portion. **T** contains at least one link, but will never have initial, final or consecutive links. The integer **C** represents the production cost for the corresponding portion ($1 \leq C \leq 1000$).

Note that in every test case, all of the genes are different from one another, and all of the portions are also different from one another. The end of the input is signalled by a line containing two times the number **-1**.

Output

For each test case, you should print a single line containing an integer number, representing the maximum net benefit that GigaFarma can obtain from a producible and alien DNA chain. If no net benefit is positive, you should print the value 0. If the net benefit can be arbitrarily large, you should print an asteris ******.

Example

Input:

4 6
hola 5
como 5
les 3
va 2
como-co 3
mo-co 8
mo-les 4
como-cómo-les 12
ta-no-sirven 100
hasta-es 200
2 3
xyz 1000
zyxyz 1000
xyz-zyx 1
zyx-xyz 1
xyz-xyz-zyx-xyz 1
2 1
abc 1
abcabc 1000
abc-abc 999
1 1
ser 10
no-ser 5
-1 -1

Output:

6
0
*
0