Problem B

Cheap B-Subsequence

Some time ago, Dejan Stojanovic, a Serbian poet, said: ‘’Words rich in meaning can be cheap in sound effects.” Is it true? A String Processing professor at UFPE wants to test this quote with strings. For that, he defined what he calls a “cheap B-subsequence”. A cheap B-subsequence, according to his definition, is a subsequence of size B, of a string S ($B \leq |S|$), that has the lowest associated cost. To define the cost of a string, the professor determined a series of rules to each letter of the alphabet. The alphabet that he used contains only lowercase letters. The rule of a letter is defined as a set of pairs (P_i,C_i), which indicates that if this letter appears in a position X on the subsequence, where X is a multiple of P_i, then the cost of $(X/P_i)*C_i$ will be added to the total cost of this subsequence. Let’s show an example. Suppose we have the following rules:

[a] = {(2,3), (4,10)}
[b] = {(1,4), (7,50)}
[c] = {(1,2), (4,20)}
[d..z] = { } // there are no rules for the characters ‘d’ to ‘z’

Suppose we have the string *abaabcbc*, and B = 4. If we choose the subsequence *aabc* (*a*b*aa*b*c*bc*)*, we would do the following procedure to calculate the associated cost:

1. The first letter of the sequence is an ‘a’, and the position 1 is neither multiple of 2 or 4, so the cost is 0;
2. The second letter of the sequence is another ‘a’, and the position 2 is a multiple of 2, so we’ll add the cost of $(\frac{2}{2})*3 = 3$;
3. The third letter of the sequence is a ‘b’, and the position 3 is multiple of 1, so we will add the cost of $(\frac{3}{1})*4 = 12$;
4. The last letter of the sequence is a ‘c’, and the position 4 is a multiple of 1 and 4, so we will add the cost of $(\frac{4}{1})*2 + (\frac{4}{4})*20 = 28$.

The total associated cost to this subsequence is 43, which is not the lowest cost, since we could have chosen aaab (*a*b*aa*b*c*bc*) and obtained an associated cost of 19 - this is indeed the cost of the cheap B-subsequence. Given the string S and the integer B, and the rules of the alphabet, your task is to create a program that tells the professor the cost of the cheap B-subsequence.

Input

The first line contains T ($T \leq 100$) – the number of test cases, after this line T test cases follows. The first line of a test case contains a string S of lowercase letters and an integer B ($1 \leq B \leq |S| \leq 100$). Each of the next 26 lines describe the rule of each letter. The first of the 26 lines corresponds to the rule of the letter ‘a’; the following line corresponds to the rule of the letter ‘b’; the last of the 26 lines corresponds to the rule of the letter ‘z’. Each line containing a rule is described in the following way: $Q\ P_1\ C_1\ P_2\ C_2 \ldots P_Q\ C_Q$ ($1 \leq Q \leq 10$; $1 \leq p_i \leq |S|$; $1 \leq c_i \leq 50$), where Q is the amount of pairs associated to this rule, and is followed by the pairs themselves.

Output

For each test case print a line containing "Case #X: Y", where X is the case number, starting at 1, and Y is the cost of the cheap B-subsequence.

Sample Input	**Sample Output**
2 abcd 1 1 1 20 1 1 15 1 1 8 1 1 30 1 1 2 0 ***(21 lines)*** abaabcbc 4 2 2 3 4 10 2 1 4 7 50 2 1 2 4 20 0 ***(23 lines)***	Case #1: 8 Case #2: 19

Author: Gustavo Stor