Round A - Kick Start 2018

Scrambled Words

PROBLEM ANALYSIS

Problem

Professor Scrmable noticed spelling mistakes in a research paper she was reviewing, but she had no difficulty in reading or understanding the words. Upon doing some research, she found an interesting article as described below:

According to a study at an English University, it doesn't matter in what order the letters in a word are, the only important thing is that the first and last letter be at the correct place. The rest can be a total mess and you can still read it without a problem. This is because the human mind does not read every letter by itself but the word as a whole.

Or rather ...

Aoccdrnig to a study at an Elingsh uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the corecrt pclae. The rset can be a toatl mses and you can sitll raed it wouthit a porbelm. Tihs is bcuseae the huamn mnid deos not raed ervey lteter by istlef, but the wrod as a wlohe.

Professor Scrmable wants to explore this concept further and starts compiling different sentences containing similarly scrambled words to send to a popular publication. Unfortunately, the space key on the professor's keyboard is not working, so she has produced one long string of characters. She has asked you to determine how many of the words in her dictionary appear (at least once) as substrings in the long string of characters, either in their original or scrambled forms. (A scrambled form consists of the same set of letters with the first and last letters in the same places, and the others in any order.)

Note that a dictionary word can appear multiple times in the string (though it should be counted only once since we only need to know whether it shows up at least once). For example, if we had the word `this` in the dictionary, the possible valid words which would be counted are `this` (original version) and `tihs` (scrambled version), whereas `tsih`, `siht` and other variations are not valid since they do not start with `t` and end with `s`. Also, `tis`, `tiss`, and `thiss` are not scrambled forms, because they are not reorderings of the original set of letters.

Since the professor is extremely busy, she gives this task to you, her favorite and most trusted research assistant. Given a dictionary, can you count the number of words in the dictionary that

appear as a substring in the professor's string at least once, in either their scrambled or original forms.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each testcase contains three lines. The first line contains an integer **L**. The second line contains a list of **L** words made of lowercase English letters; these make up the dictionary. The third line contains two lowercase English letters $\mathbf{S_1}$ and $\mathbf{S_2}$, and five integers **N**, **A**, **B**, **C** and **D**. $\mathbf{S_1}$ and $\mathbf{S_2}$ are the first two characters of the professor's string **S**, **N** is the length of **S**, and the other four integers are parameters that you should use to generate the characters of **S**, as follows:

First we define ord(c) as the decimal value of a character `c` and char(n) as the character value of a decimal n. For example, ord('a') = 97 and char(97) = 'a'. You can refer to ASCII table for other conversions.

Now, define x_1 = ord($\mathbf{S_1}$), x_2 = ord($\mathbf{S_2}$). Then, use the recurrence below to generate x_i for i = 3 to **N**:

- x_i = (**A** * x_{i-1} + **B** * x_{i-2} + **C**) modulo **D**.

We define $\mathbf{S_i}$ = char(97 + (x_i modulo 26)), for all i = 3 to **N**.

Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the number of words from the dictionary that appear (in their original or scrambled forms, as defined above) as substrings of the given string.

Limits

$1 \le \mathbf{T} \le 20$.
Memory limit: 1 GB.
No two words in the dictionary are the same.
Each word in the dictionary is between 2 and 10^5 letters long, inclusive.
The sum of lengths of all words in the dictionary does not exceed 10^5.
$\mathbf{S_1}$ and $\mathbf{S_2}$ are lowercase English letters.
$0 \le \mathbf{A} \le 10^9$.
$0 \le \mathbf{B} \le 10^9$.
$0 \le \mathbf{C} \le 10^9$.
$1 \le \mathbf{D} \le 10^9$.

Small dataset (Test set 1 - Visible)

Time limit: 20 seconds.
$1 \le \mathbf{L} \le 1000$.
$2 \le \mathbf{N} \le 1000$.

Large dataset (Test set 2 - Hidden)

Time limit: 150 seconds.
$1 \leq \mathbf{L} \leq 20000$.
$2 \leq \mathbf{N} \leq 10^6$.

Sample

Sample Input	Sample Output
1 5 axpaj apxaj dnrbt pjxdn abd a a 50 1 1 1 30	Case #1: 4

In Sample Case #1, using the generation method, the generated string **S** is `aapxjdnrbtvldptfzbbdbbzxtndrvjblnzjfpvhdhhpxjdnrbt`. Scrambled or original occurences of dictionary words are highlighted as follows:

- `axpaj` occurs in its scrambled form as `aapxjdnrbtvldptfzbbdbbzxtndrvjblnzjfpvhdhhpxjdnrbt`.
- `apxaj` occurs in its scrambled form as `aapxjdnrbtvldptfzbbdbbzxtndrvjblnzjfpvhdhhpxjdnrbt`. Note that even though `apxaj` is the scrambled form of another dictionary word `axpaj`, both should be counted.
- `dnrbt` occurs twice in its original form as `aapxjdnrbtvldptfzbbdbbzxtndrvjblnzjfpvhdhhpxjdnrbt`, though it should be counted only once.
- `pjxdn` occurs in its scrambled form as `aapxjdnrbtvldptfzbbdbbzxtndrvjblnzjfpvhdhhpxjdnrbt`. Note this occurence overlaps with occurence of another dictionary word, but still they're counted independently.
- `abd` doesn't occur at all.

Note: We do not recommend using interpreted/slower languages for the Large dataset of this problem.