

Problem A

Soundex

Soundex is a phonetic algorithm for transforming a string into a code, which is always a letter followed by three digits. The purpose is that strings with a different spelling but similar pronunciation will be transformed into the same code. The transformation rules are as follows:

- The first letter of the string is the first letter of the code.
- Subsequent consonants are replaced by digits:
 - b, f, p, v with 1
 - c, g, j, k, q, s, x, z with 2
 - d, t with 3
 - l with 4
 - m, n with 5
 - r with 6

h and w are ignored. The vowels, a, e, i, o, u and y, are not encoded.

- Two or more adjacent letters with the same digit are replaced with a single digit. Two or more letters with the same digit separated with h or w are also replaced by a single digit. Two letters with the same digit separated by a vowel will appear as the digit twice.
- Repeat the previous step until it isn't possible to replace repeating digits with one digit.
- If the resulting code has less than 3 digits, pad the end with zeroes until the code has 3 digits. If there are more than 3 digits, drop the rightmost digits.

Some example transformations are:

- Both **robert** and **rupert** are transformed to R163.
- **baawwww** is transformed to B000.
- **hopp** is transformed to H100. The first letter of the string always becomes the first letter in the code, even if it is a vowel or h or w.
- **ratatata** is transformed to R333, because the vowel between each pair of t (3) forces the digit to be repeated.
- **yhhhwhthwhthwhthwhth** is transformed to X300. All occurrences of h and w are ignored, leaving only one 3 in the code.
- **bbpb** is transformed to B100. The first b is kept separate from the last three b's. The remaining b's are consecutive and are replaced with one digit.



Your evil friend Halvor has noticed that a lot of different words will give the same soundex code. For example, the strings `rhhhbm`, `rubeno`, `rpowam`, `robnew` and `73908` other strings with 6 or less letters will be converted to the code `R150`. This makes him very curious, and he wants to know the number of strings of the a given length or shorter that will be converted to the same soundex code. Naturally, he wants you to write a computer program to accomplish this task. Your friend does not care about upper or lower case, so `AA`, `Aa`, `aa` etc are considered to be equal strings and should only be counted once.

Input specifications

The first line of input contains a single integer T , the number of test cases to follow. Each test case begins with a line containing a string S and an integer number L . S represents a soundex code, and consists of one uppercase letter followed by three digits. L represents the maximum length of the original string which is converted into the given soundex code.

Output specifications

For each test case output one line containing a single number, the number of strings of length L or less having the soundex code S . This number can be large, so output it modulo 1,000,000,007.

Notes and Constraints

- $0 < T \leq 100$
- $0 < L \leq 1000$
- All soundex codes will start with an uppercase letter.
- Two strings S and T are equal if they have the same length and the letters s_i, t_i at each position are equal, regardless of case.
- Only letters between `a` and `z` are considered. No `æ`, `ø`, `å` or other non-English characters are to be used.
- The soundex code is always legal. That is, a non-zero digit will never follow a zero digit, and the digits range from 0 to 6.

Sample input

```
3
A300 2
R150 6
X123 1
```

Output for sample input

```
2
73912
0
```