7532 In a planet far far away

CodeChef has sent Jalebi Bai to a new planet to learn a special recipe from them. On her journey from the space station to the head chef's house, she observed that the multiplication rule on this new planet is slightly different from our Earth.

For example: on this planet, $2 * 4$ may be 9 instead of 8. Weird, she thought.

Curious, Jalebi Bai asked a copassenger the results of all the 100 multiplications between all pairs of digits.

She figured out that the Standard algorithm for multiplication on this planet follows the same rules as on Earth except that digit multiplication should follow the rule of that planet.

She also found out that any digit multiplied with 0 is 0, the multiplication of x with y is the same as the multiplication of y with x.

Addition is defined to be same as that on the Earth.

For example, if the result of 2 multiplied with 4 is 9, 5 multiplied with 3 is 19, 2 multiplied with 3 is 17 and 4 multiplied with 5 is 24, then multiplication of 25 with 43 will give:

```
  25
  43
----
  19
 170
 240
 900
----
1329
```

The final answer will be 1329 (following the addition algorithm of the earth for managing carry).

Finally Jalebi Bai meets the head chef of the planet, but the problem is he is ready to give her the recipe only if she can solve a puzzle. He asks her to find the sum of the results of multiplication (according to the planet) between all possible ordered pairs of numbers from the set of positive integer less than or equal to a given number A. **Note:** If i and j are distinct, then the pair (i, j) is considered different from (j, i).

Jalebi Bai then asks him for a formal multiplication algorithm of the planet, to which the head chef agrees and gives following description.

The multiplication of two numbers X and Y represented as arrays, with M as the digit multiplication matrix, is defined as follows:

```
Multiply(X[1..p], Y[1..q])
{
    m = [1..p+q]  //Allocate space for result
    for b = 1 to q
    {
        carry = 0
        for a = 1 to p
        {
            m[a + b  1] += carry + M[X[a]] [Y[b]]
            carry = m[a + b  1] /10
```

```
            m[a + b  1] = m[a + b  1] mod 10
        }
        m[b + p] += carry
    }
    answer = 0
    for r = p+q to 1
        answer = answer*10 + m[r]
    return answer
}
```

Aliens cannot hear any large number, so they want to hear the output modulo Mod, where Mod is some integer.

If Jalebi Bai gives a wrong output, she would be eaten up! (Surprise surprise!) Jalebi Bai can contact someone on earth for help. She reaches you for a code which can evaluate the output for any given input. Can you help her solve this puzzle?

Input

The input file contains several test cases, each of them as described below.

The digit multiplication rule appears in 9 lines. Line i contains the multiplication rule for the digit i. Each line contains 9 positive integers, each of which is less than 100. It is guaranteed that the given matrix is symmetric.

The next line contains T, the number of puzzles to evaluate.

The first line of each puzzle contains a positive integer A.

The second line of each puzzle contains Mod.

Output

For each test case and puzzle, output the solution modulo Mod in a single line.

Constraints:

- $1 \leq T \leq 50000$
- $1 \leq A < 10^{100000}$
- $1 \leq Mod \leq 10^8$
- $1 \leq$ Sum of lengths of all strings $A \leq 5 * 10^5$

Explanation:

Example case 1. Normal multiplication rule follows.

Example case 2. The only pairs with multiplication different from normal multiplication are (1, 2), (1, 3), ..., (1, 9) and (2, 1), (3, 1), ..., (9, 1).

Sample Input

```
1 2 3 4 5 6 7 8 9
2 4 6 8 10 12 14 16 18
3 6 9 12 15 18 21 24 27
4 8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
```

```
7 14 21 28 35 42 49 56 63
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
3
9
10000
73
100
83
100
1 1 1 1 1 1 1 1 1
1 4 6 8 10 12 14 16 18
1 6 9 12 15 18 21 24 27
1 8 12 16 20 24 28 32 36
1 10 15 20 25 30 35 40 45
1 12 18 24 30 36 42 48 54
1 14 21 28 35 42 49 56 63
1 16 24 32 40 48 56 64 72
1 18 27 36 45 54 63 72 81
1
9
10000
```

Sample Output

```
2025
1
96
1953
```