A good encoding program has the following properties.

- it has some different symbols.
- Each of these different symbols is encoded into different strings which contains digits from 0 to $k-1$. Like for 3 different symbol and $k=2$ corresponding encoding code can be 0,10,11.
- You can encode a string containing this different symbol by just concatenating their corresponding encoding code. Like from the previous example the encoding of the string babc is 1001011.
- You select the encoding of these symbols in such a way that you can decode the encoded string without any ambiguity. Means if you build a prefix tree with these encoding code then each of the node will have either k child or none. Huffman tree is a good example with similar tree $k=2$.

Now you have a set of $n+m$ different symbol. But you have lost the encoding string of m of those. Given the encoding code of the rest of the n symbols you have determine how many ways you can select the encoding set of the lost m symbols.

Input

First line contains T ($1 \leq T \leq 100$) the number of test cases. Then T test cases follow.

First line of each test case contain 3 integer n ($0 \leq n \leq 1000$),m ($1 \leq m \leq 200$) and k ($2 \leq k \leq 5$). Each of the next n line contains a string containing digits from 0 to $k-1$. This is encoding code for a symbol. These n codes are valid. Means none of these n string will not be prefix of one another.

Output

For each test case find the number of way you can select the other m encoding string set. The number of way may be huge. Output the result%10007.

Sample Input

```
5
0 5 2
0 5 3
1 5 2
000
2 4 2
01
10
3 20 3
012
120
201
```

Sample Output

```
14
3
9
5
5313
```