Problem UTF8SEARCH: UTF-8 Search

UTF-8 was invented to be able to encode more characters. But it makes it also possible to have many encodings for the same character. There are some people, who think that they can use this to protected their copyright. Because document viewers are able to handle whole UTF-8 but many printers can't print all UTF-8. So when you use sparse used encodings of a character, you can see the right character in the viewers, but you can't print it. Peter has some of these documents and he doesn't want to break copyrights but he wants to search in these documents. But the search of the viewers search only exact matches and can't handle same characters with different encoding as one character. So Peter ask you to write a program, that can find ascii strings in multi byte encoded text.
The program should not only handle UTF-8, but many encodings, which consists of one and two byte characters. These encodings have the following properties:

- Only characters looking like ascii characters are relevant.
- Normally a character is encoded by one byte.
- A one byte character normally looks like it would look like in ascii.
- A two byte character normally looks like its second byte as ascii.
- If there is one two byte character starting with byte b all characters starting with b are two byte characters.

Input

In the first line the number of test cases is given. At the begin of each test case the encoding is described. First the number of non standard looking characters is given. For every of those characters a line is given with the form $s = c$, where s is a string of one or two bytes, which encodes this character and c is the ascii character, which looks like the character.
After the encodings a line with the text in the given encoding is given. This text contains at most 1000 letters. The last line of the input contains the search string in ascii. The search string consists of at most 100 letters.
The decoded text in the sample input is "abba".
For simplicity all used characters in the input are alphanumeric or spaces.

Output

Output a line for every test case. Output "found at x", where x is the position in the text where the search string starts to match first, when you can find the search string. Otherwise output "not found".
Sample Input and Output are provided on the next page.

Sample Input 1

```
2
7
a0=a
a1=a
a2=a
b0=b
b1=b
b2=b
c=a
a0b1bbc
abba
0
A ascii text
ascii
```

Sample Output 1

```
found at 1
found at 3
```