2601 - Touchscreen Keyboard

Description

Nowadays, people do not use hardware keyboards but touchscreens. Usually, they touch on the wrong letters with their chunky fingers, because screen space is precious and the letters therefore too small.

Usually, a spell checker runs after typing a word and suggests other words to select the correct spelling from. Your job is to order that list so that more likely words are on top.
The typical touchscreen keyboard looks like this:

qwertyuiop
asdfghjkl
zxcvbnm

You should use the distance between the letters to type a word: the distance is the sum of the horizontal and vertical distance between the typed and proposed letter. Assume you typed a **w**, the distance to **e** is 1, while the distance to **z** is 3.

The typed word and the list of words from the spell checker all have the same length. The distance between two words is the sum of the letter distances. So the distance between **ifpv** and **icpc** is 3.

Input specification

The first line of the input specifies the number of test cases **t (0 < t < 20)**. Each test case starts with a string and an integer **l** on one line. The string gives the word that was typed using the touchscreen keyboard, while **l** specifies the number of entries in the spell checker list **(0 < l 10)**. Then follow **l** lines, each with one word of the spell checker list. You may safely assume that all words of one test case have the same length and no word is longer than 10 000 characters (only lowercase **'a' - 'z'**). Furthermore, each word appears exactly once in the spell checker list on one test case.

Output specification

For each test case, print the list of words sorted by their distance ascending. If two words have the same distance, sort them alphabetically. Print the distance of each word in the same line.

Sample input

```
2
ifpv 3
```

```
iopc
icpc
gcpc
edc 5
wsx
edc
rfv
plm
qed
```

Sample output

```
icpc 3
gcpc 7
iopc 7
edc 0
rfv 3
wsx 3
qed 4
plm 17
```

Hint(s)

Source	German Collegiate Programming Contest 2012
Added by	**ralcolea**
Addition date	2013-11-01
Time limit (ms)	20000
Test limit (ms)	1000
Memory limit (kb)	65535
Output limit (mb)	64
Size limit (bytes)	30000
Enabled languages	Bash C C# C++ Java Pascal Perl PHP Python Ruby Text