You’ve been tasked with analyzing messages to look for various threats; however, the messages have been encrypted. Luckily, they’re only encrypted using a Caesar cipher, which means that each letter in the plaintext is shifted a fixed number of positions later in the alphabet. The alphabet wraps around, so after z is a, and after Z is A. For instance, the message “how are you?” with a shift of 5 would become “mzb fvj dzr?”. To decrypt a message, you simply shift the letters in the opposite direction.

For each message, you will need to determine the key (shift value) and the plaintext corresponding to the given ciphertext. To assist you, you will be given lists of known threat words and known non-threat words. Matching against words in these lists is not case sensitive, e.g. “word” matches “Word”.

You can assume you guessed the correct key if it results in more known words (threat and non-threat) than any other key. For the purpose of this task, a word is an uninterrupted sequence of 1 ≤ x ≤ 20 characters A-Z and/or a-z. All other characters are not effected by the shift and are the same in the plaintext and the ciphertext. All messages (both plaintext and ciphertext) consist only of printable characters, including space.

Input
The first line of input will contain the number of test cases, C (1 ≤ C ≤ 50). Each test case will begin with a line containing the number of known non-threat words, G (1 ≤ G ≤ 50). The following G lines will each contain a lower-case non-threat word. The next line will contain the number of known threat words, B (1 ≤ B ≤ 50). The following B lines will each contain a lower-case threat word. The next line will contain a message of 1 ≤ y ≤ 1000 characters of ciphertext. No line will contain any leading or trailing whitespace.

Output
Each test case will produce one or two lines of output.
If the key and plaintext can be determined, the first line of output should contain the plaintext message, preserving the case and punctuation of the original ciphertext, and the second line should contain ‘Shift: S, Match: M%, Threat: T%’, where S (0 ≤ S < 26) is the number of characters the plaintext was shifted to encrypt the message, M is an integer indicating what percentage of the words in the plaintext are known words (threat and non-threat), and T is an integer indicating what percentage of the words in the plaintext are threat words. All percentages should be rounded to the nearest 1%.
If the key and plaintext cannot be determined, there should be a single line of output containing ‘Unable to decipher’.

Sample Input
2
11
a also from if it keep must to want you yourself
2
hide secret
Vs lbh jnag gb xrq c n frqerg, lbh zhfg nyfb uvqr vg sebz lbhefrys.
12
a a brother darkness destruction is it language mind of the thing words 3
beautiful freedom truth
Ha’r z adztshek sgghf, sgd cdrgtbsbm ne vnqcr.

Sample Output
If you want to keep a secret, you must also hide it from yourself.
Shift: 13, Match: 100%, Threat: 14%
It’s a beautiful thing, the destruction of words.
Shift: 25, Match: 89%, Threat: 11%