ACM-ICPC Live Archive

3727 The Mastermind Master's Mind

The Advancement of Collegiate Mastermind (hey, that's ACM again...weird!) is an organization which (among other things) holds classes for college students to improve their Mastermind skills. Recall that basic Mastermind is a two-player game which works as follows: The first player — the codemaker — creates a secret 4-color code, each color taken from one of six colors (we'll use A,B,C,D,E and F for the colors). The other player — the codebreaker — now must try to guess the codemaker's code. After each guess, the codemaker uses black and white pegs to tell the codebreaker two things: the number of correct colors in the correct positions (the black pegs), and the number of remaining correct colors that are in the wrong positions (the white pegs). For example, if the true code is ABCC, and the codebreaker makes the guess ACCD, then the response would be 2 black and 1 white; if the guess was CCAA, the response would be 3 white. The codebreaker continues making guesses until he receives 4 blacks. More advanced versions of Mastermind increase both the length of the code and the number of colors to choose from.

The ACM's master instructor is Dee Sifer, and she has a unique ability: when given a set of n guesses and responses, she can immediately determine what the best $(n+1)$-st guess should be. For each possible $(n+1)$-st guess, Dee calculates (in her head) the number of codes left for each possible response she could get to that guess. The maximum of these numbers over all responses is called the *uncertainty* of the guess. The "best" guess is the one with the minimum uncertainty. For example, suppose that you get to a point in a game where you've narrowed down the answer to only three possible codes: ABBB, ABBC or ABCB. If your next guess is ABBB, there would be two possible responses: 4 blacks (leaving 0 remaining possibilities left) or 3 blacks (leaving 2 remaining possibilities — ABBC and ABCB). However, if instead of ABBB you try ABBC, then there are 3 possible responses: 4 blacks (leaving 0 possibilities), 3 blacks (leaving 1 possibility — ABBB) and 2 blacks and 2 whites (also leaving 1 possibility — ABCB). Thus ABBC would be a better guess in this case, since the uncertainty it leaves is 1, whereas the uncertainty for ABBB is 2.

This is all well and good, except for one thing. You have been selected as Dee's successor, and you do NOT have Dee's ability to pick the minimum uncertainty guess. Dee has been dropping hints (in code) that she plans to retire soon, so you need a program to help you simulate her ability.

Input

Input will consist of multiple test cases. The first line of the input file will contain a single integer indicating the number of test cases. Each test case will consist of several lines. The first line will contain three integers: 'l c n', where l is the length of the code being guessed, c is the number of colors to choose from, and n is the number of guesses made so far. These values will satisfy $1 \le l \le 15$, $1 \le c \le 20$, $0 \le n \le 10$. The values of l and c will be such that the total possible number of codes will be ≤ 32768. After this will come n lines of the form

guess b w

where *guess* is a length-l character string specifying a guess, and b and w are the number of black and white pegs in the response. All colors will be uppercase letters taken from the first c letters of the alphabet. For each test case, the guesses given will limit the total number of possible solutions to ≤ 1500.

Output

For each test case, output a single line containing the best guess and its uncertainty. Use a single blank to separate the guess from the uncertainty. If there is more than one guess with the same minimum uncertainty, use the one which comes first lexicographically.

Sample Input

```
3
4 6 2
AABC 1 2
BEAC 0 3
4 6 1
ABCD 0 0
3 20 4
ABE 1 0
ROM 1 0
INK 1 0
MOB 0 2
```

Sample Output

```
ABCD 4
AEEE 3
IBM 0
```