



acm International Collegiate
Programming Contest

2003



event
sponsor

ACM International Collegiate Programming Contest 2003

South America

Warm Up Session, 7th November 2003

(This problem set contains two problems; pages are numbered from 1 to 3)

Hosted by

Universidad de Buenos Aires (UBA), Buenos Aires, Argentina

Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil

Universidade de Fortaleza (UNIFOR), Fortaleza, Brazil

Universidade Federal de Mato Grosso do Sul (UFMS), Campo Grande, Brazil

Universidade Regional Integrada do Alto Uruguai e das Missões(URI), Erechim, Brazil

Universidad del Bío-Bío, Chile

Universidad Dr. Rafael Belloso Chacín, Maracaibo, Venezuela

Problem A

Divide and Conquer

Source file: divide.pas, divide.c, divide.cc or divide.java

Given two integers M and N , with $1 \leq M \leq N \leq 5000$, you must determine the integer numbers X and Y such that

- A. $M \leq X \leq N$; and
- B. Y is the number of divisors of X ; and
- C. Y is the largest possible; and
- D. X is the largest possible.

Input

Your program should process several test cases. Each test case is composed by a single line containing the two integers M and N ($1 \leq M \leq N \leq 5000$). The end of input is indicated by $M = N = 0$.

The input must be read from standard input.

Output

For each test case in the input your program should print a single line, containing the two integers X and Y , separated by a space character.

The output must be written to standard output.

Sample input

Output for the sample input

1 5	4 3
300 500	480 24
4500 5000	4680 48
0 0	

Problem B

Make a Guess

Source file: guess.pas, guess.c, guess.cc or guess.java

MasterMind is a game that was created by a telecommunication engineer named Mordechai Meirovitz, from Romania, in 1971. The game was awarded the prestigious “The Year’s Game” in 1974 and was a huge commercial success, having been sold in more than 40 countries.

MasterMind is a logical game in which your objective as a player is to find out a secret password chosen by an opponent player. The password to be found is a sequence of characters from a given alphabet. In order to find out what the password is, you submit “guesses” to your opponent. A guess is a possible password. That is, a guess is a sequence of the same number of characters as the password, from the same alphabet.

After each guess you receive an answer, which consists of two integers, (E, G) , indicating how good your guess was. If a character in a guess exists in the password at the same position (in the string) then it counts as “excellent” (E). Otherwise if a character exists in both the guess and the password, but in a different position in the string, then it counts as “good” (G). A single character from a guess is not counted twice (that is, if it is counted as excellent it is not counted as good). The table below illustrates some examples.

Password	Guess	Answer	Remark
1233	3243	(2, 1)	‘2’ and the second ‘3’ are at the same position in both the password and guess, and because the first ‘3’ also exists in the password, but in another position.
1233	3000	(0, 1)	The only coincident character in both the password and the guess is the ‘3’, but in different positions.
1233	4455	(0, 0)	‘4’ and ‘5’ are not present in the password.
1233	1233	(4, 0)	Right!

Based on the answers received after each guess, a player can eventually deduce what the password is. The goal of the game is to find out the password with the minimum number of guesses.

You must write a program to help you to find the password, showing all deduced positions after a set of guesses and answers.

Input

The input contains several test cases. The first line of a test case contains three integers N , L and K . N indicates the number of guesses ($0 \leq N \leq 1000$), L indicates the length of the password ($1 \leq L \leq 4$) and K indicates the size of the alphabet used ($1 \leq K \leq 18$). The next line contains K characters that can be digits (‘0’ to ‘9’) or capital letters (‘A’ to ‘Z’), specifying the alphabet used in the password. The following N lines contain each a guess and an answer. A guess is a sequence of L characters from the

alphabet used. An answer is composed by two integers E and G , (respectively the excellency and goodness of the guess, as explained above). The end of input is indicated by $N = L = K = 0$.

The input must be read from standard input.

Output

For each test case in the input file your program must print one single line containing the information deduced from the set of guesses and answers. If it is possible to deduce the entire password from the set of guesses and answers, then your program should print the password. Otherwise, it should print a character ‘?’ in each non-deduced position of the password, as shown in the examples.

The output must be written to standard output.

Sample input

Output for the sample input

3 4 2	?B?B
AB	2212
AAAA 1 0	??
BBBB 3 0	
ABAB 3 0	
4 4 2	
12	
1111 1 0	
2222 3 0	
1212 3 0	
2211 3 0	
1 2 3	
123	
32 1 1	
0 0 0	