Problem A. FizzBuzz V3

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

The FizzBuzz problem is a problem given by interviewer to filter out programmers who can't program. The question is simple, list down integers from 1 to 100, but if the number is divisible by 3, print "Fizz" instead. If the number is divisible by 5, print "Buzz" instead. If the number is divisible by both 3 and 5, print "FizzBuzz".

If you are like me, then you must feel some serious itch with this question. Why must it be from 1 to 100 only? Why must it be 3 and 5? Why must it be "Fizz" and "Buzz". Why only two integer? It feels so 'hardcoded' and not future proof. In actual development, it is normal and a good idea to just hardcode such simple problem rather than making a complex and error prone solution. However, this is not actual development.

Your task is to make a future proof solution for the FizzBuzz problem. Instead of from 1 to 100, you are to print from a to b (which a can be higher than b). And instead of only 3 and 5 and "Fizz" and "Buzz" , you are given n integers d_i and n string s_i.

Your task is to print integers, each in its own line, from a to b. If the integer is divisible by the integer d_i, print s_i. If the integer is divisible by more than one d_i (where $1 \leq i \leq n$) print all the corresponding e_i with no space between them, in order which they are inputted. There can be e_i which is equal to e_j where $i \neq j$.

Input

The first line consist of three integer a, b, n $(1 \leq a, b \leq 10^5)$ $(1 \leq n \leq 100)$.

The next n line consist of an integer and a string d_i, s_i $(1 \leq d_i \leq 10^5)$. s_i does not contain any spaces. s_i will not exceed 10 character.

Output

Print integers from a to b in its own line.

If it is divisible by any integer d_i $(1 \leq i \leq n)$, then print the corresponding s_i.

If it is divisible by more than one d_i, print all s_i in order given in the input without space between them. (See example 2).

Examples

standard input	standard output
9 16 2 3 Fizz 5 Buzz	Fizz Buzz 11 Fizz 13 14 FizzBuzz 16
20 10 4 3 Ketigaan 2 Keduaan 5 Kelimaan 2 DuaLagi	KeduaanKelimaanDuaLagi 19 KetigaanKeduaanDuaLagi 17 KeduaanDuaLagi KetigaanKelimaan KeduaanDuaLagi 13 KetigaanKeduaanDuaLagi 11 KeduaanKelimaanDuaLagi

Note

In example two, notice how the divisor 2 is repeated and the order it is given in the input matters in the output.