Fizzy Buzzy Bears (prob3)

The Problem

FizzBuzz is a popular programming exercise that is often used in job interviews (really, see http://en.wikipedia.org/wiki/Fizz_buzz) . But you're at a programming contest at Mercer University, not in a job interview, so instead of the classic FizzBuzz, you'll do Fizzy Buzzy Bears.

To play Fizzy Buzzy Bears, you'll print out a series of integers from a given low to a given high bound, except when the numbers are divisible by special values:

- If the number is divisible by 3 (the number of main Mercer campuses), print "Mercer" instead of the number.
- If the number is divisible by 4 (the number of Mercer Regional Academic Centers), print "Bears" instead of the number.
- If the number is divisible by 18 (Mercer was founded in 1833), print "Grrrr" (there are 4 r's) instead of the number.
- If the number is divisible by 33 (Mercer was founded in 1833), print "Bite" instead of the number.

If a number is divisible by more than 1 of 3, 4, 18, or 33, print all the applicable words in order of the divisors, so 12 should print "MercerBears", 36 should print "MercerBearsGrrrr", and 66 should print "MercerBite".

Input

The input will consist of one or more input sets. Each set will have two integers, a low and high bound. The end of input will be an input set with a low bound that is greater than the high bound. This last case should not be processed.

Output

For each input set, have a line with the case number (where the first case is numbered 1 and the others are numbered sequentially). Then list the integers from the lower bound to the upper bound, inclusive, with the Fizzy Buzzy Bear value if there is one or the integer, one per line.

Have a blank line after the output for each input set.

Sample Input

```
1 10
5 20
30 35
10 1
```

Sample Output

```
Case 1:
1
2
Mercer
Bears
5
Mercer
7
Bears
Mercer
10

Case 2:
5
Mercer
7
Bears
Mercer
10
11
MercerBears
13
14
Mercer
Bears
17
MercerGrrrr
19
Bears

Case 3:
Mercer
31
Bears
MercerBite
34
35
```