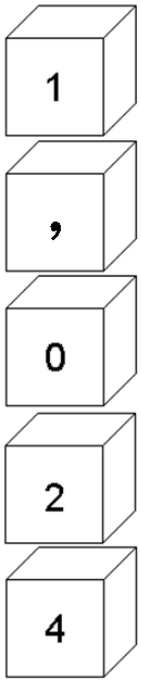
University of Michigan-Dearborn
Fall 2009 CIS Programming Contest
September 25, 2009

Sponsored by:
Upsilon Pi Epsilon Student Organization
The Department of Computer and Information Science

Rules:

1. There are five questions to be completed in three hours.
2. All questions require you to read the test data from standard input and write results to standard output. You cannot use files for input or output. Additional input and output specifications can be found in the General Information Sheet.
3. The allowed programming languages are C, C++, C# and Java.
4. All programs will be re-compiled prior to testing with the judges' data.
5. Non-standard libraries cannot be used in your solutions. The Standard Template Library (STL) and C++ string libraries are allowed. The standard Java API is available, except for those packages that are deemed dangerous by contestant officials (e.g., that might generate a security violation).
6. Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required.
7. All communication with the judges will be handled by the PC2 environment.
8. Judges' decisions are to be considered final. No cheating will be tolerated.

A. Power Tower



Snoopy loves going to Cedar Point to look at roller coasters. He imagines these wonderful structures as a series of big cubes stacked one onto another. Snoopy also loves to multiply numbers by 2. When multiplying numbers, Snoopy imagines them as big rollercoaster towers with stacked cubes. Each cube has a digit on it, representing a part of the number. For example, if you take the number one and multiply it by 2 ten times, you will get the number 1,024. Snoopy likes to imagine commas between each group of 3 digits, starting from the bottom. So, for 1,024 he imagines a tower of five cubes, where the topmost cube has the digit 1, the next one under it has the comma, and the last cube on the bottom has the digit 4.

Snoopy likes really tall coaster towers, but he has trouble imagining them. Snoopy wants to see a tower that has fifty, a hundred or even more cubes. Such towers happen when he multiplies one by 2 many many times. Snoopy knows that you like numbers and wants you to help him visualize these tall and beautiful towers. Can you help Snoopy?

Input

Input will consist of multiple cases. Each line will contain an integer n, where $1 \leq n \leq 1000$. The last line of input will have number 0 and should not be processed.

Output

For each test case output the number you get when you raise 2 to the n^{th} power. For each number, separate each set of 3 digits by commas, starting from the right of the number.

Sample Input

```
4
10
20
0
```

Sample Output

```
16
1,024
1,048,576
```

B. The Broken Record of DJ Zhzyatslya

DJ Zhzyatslya is in trouble. You see, after a successful career as a DJ, Zhzyatslya decided to try his hand at computing. But that did not work so well. After failing his Final Exam on History of Computing and Computation of Computable Things using Computers, DJ Zhzyatslya got very upset and smashed up his latest favorite one of a kind record collection. After he came back to sanity, he saw his life in pieces in front of him. Will you be so kind and help him put his records and his life back together?

By the way, DJ Zhzyatslya is very fond of his name. To ensure you pronounce it correctly, he provided you with IPA (International Phonetic Alphabet) transliteration: [ʐ ʐ ʲa ʧ l ʲa].

Input

Input will consist of multiple test cases. Each line will be in format A / B + C / D, where the letters represent numbers between 1 and 1000 inclusively. The last line will have all zeroes in place of the numbers. That line will not be processed.

Output

Sum up the fractions of DJ Zhzyatslya's Record Collection and print them out in a format A / B, if the sum is less than one. Use the format A and B / C, if the sum is larger than one. If the sum turns out to be a whole number, output only the whole part without the fraction, in a format of A. Always reduce the fractions to their lowest terms, when possible.

Sample Input

```
1 / 1 + 2 / 2
1 / 1 + 2 / 3
1 / 3 + 1 / 3
0 / 0 + 0 / 0
```

Sample Output

```
2
1 and 2 / 3
2 / 3
```

C. Not a Composite Grid

Creatively Informed Scientists (CIS) consortium is in the process of developing new grid-based games to compete with Sudoku. Matt "This Cool", lead designer of CIS and head of the Universal Program Engineering department (UPE) is a creator of the game in which he needs your help in generating solutions.

Matt "This Cool" will give you a number between 1 and 100, which he puts in the top left corner of the grid. Your task is to fill the rest of the grid with the numbers according to these rules: each number in the grid must be unique, each consecutive pair of numbers must sum to a non-composite number both vertically and horizontally, and each cell must contain the smallest number possible in each cell, starting from the top to bottom and left to right. Note that this way there are many grids possible. Matt This Cool" wants you to pick the one that's first lexicographically. This means that if there are two grids, pick one that has the least first square (1st row 1st column). If there is a tie, pick one that has the least second square (1st row 2nd column), and so on. When the first row is over, we use the second row and so on.

Input

Input will consist of multiple test cases. Each test case consists of an integer number n, $(1 \leq n \leq 100)$ that will go in the top left corner of the grid to get you started. Fill in the rest of the grid according to the problem statement above.

Output

For each test case, output a line "`Matt, here is puzzle #`i`:`", where i is the number of the puzzle, starting with 1, followed by the print out of the grid. Each grid consists of six rows and six columns of numbers, where each pair of numbers is separated by a space. There will be no spaces after the last number. For visual clarity, separate each grid print-out by a single blank line. Do not leave any extra blank lines after the last grid print-out. See Sample Output for an example.

If you have more than one grid satisfying the rules, pick the lexicographically first one.

Sample Input

```
1
23
0
```

Sample Output

Sample showing visible symbols	**Sample showing end-of-line symbols**
Matt, here is puzzle #1: 1 2 3 4 7 6 10 21 16 13 24 5 19 22 25 18 23 14 12 31 36 35 8 15 29 30 17 26 33 28 32 11 20 27 34 9 Matt, here is puzzle #2: 23 6 1 2 3 4 18 5 12 11 8 9 13 24 35 26 15 22 30 29 32 21 16 7 17 14 27 10 31 36 20 33 34 19 28 25	Matt, here is puzzle #1:\n 1 2 3 4 7 6\n 10 21 16 13 24 5\n 19 22 25 18 23 14\n 12 31 36 35 8 15\n 29 30 17 26 33 28\n 32 11 20 27 34 9\n \n Matt, here is puzzle #2:\n 23 6 1 2 3 4\n 18 5 12 11 8 9\n 13 24 35 26 15 22\n 30 29 32 21 16 7\n 17 14 27 10 31 36\n 20 33 34 19 28 25\n

D. Dogs with Large Eyes

Hans, a little boy, had many dogs. Some of his dogs had eyes as big as tea saucers. Others had eyes as big as windmills, and some of the ferocious dogs had eyes as big as skyscraper towers. Recently Hans befriended another boy named Andersen. Andersen had a lot of dogs and each dog had eyes of different sizes. Some dog's eyes were as big as electric scooters, some had eyes as big as Suzuki motorcycles, and some had eyes as large as rotating yellow Ferraris. Hans realize that some of these dogs had eyes of identical sizes. He also realized that the dogs with the same eyes had magical powers when they stood next to each other. Hans just had to know how many magical dogs he and Andersen had together. Since the dogs are wild and ferocious, they are running all over Hans' yard. Can you locate the dogs with identical eye sizes and bring them back to Hans to let the magic happen?

Input

Each input line will begin with a number n. You will be given a list of n integer numbers 1 through 100 inclusively, where the number represents the size of dog's eyes in alens. Alen is a Danish unit of length equaling to 62.77 centimeters. Hans and Andersen together will not have more than 200 dogs. The last line will contain the number 0 by itself, indicating the end of input and should not be processed.

Output

For each line of input, you will find all dogs with the same eye sizes and print out the sizes in sorted order. If you cannot find any dogs with identical eyes, print out the message "I want more dogs with large eyes!" on a line by itself.

Sample Input

```
5 1 2 3 2 4
8 1 2 2 3 3 4 4 5
5 2 1 3 4 5
0
```

Sample Output

```
2
2 3 4
I want more dogs with large eyes!
```

E. Reverse Primes

Your friend, Moe Knee, enjoys two things in life -- playing the lottery and prime numbers. He likes prime numbers so much that he plays the lottery by only betting on prime numbers. So far, he has been using any old prime numbers and his luck at winning has not been very good. He has decided to start betting only on reversible primes, hoping that his luck will improve. A reversible prime is a prime number whose 'reverse' is also a prime. To reverse a number, write it down on paper, and then write it backwards. For example, reverse of 12345 is 54321.

For instance one such reversible prime is 17, since its reversal, 71 is also a prime. Another example is 13, as 31 is also prime. Note that while 41 is prime, 14 is not!

Moe Knee quickly found out that calculating these numbers takes too long. You see, Mr. Knee Moe uses pencil and paper, so he doesn't get very far and quickly gets tired. He has asked that you help him to generate a list of reversible primes within a given range of numbers that he will be using to bet on this week.

Input

Input will consist of multiple test cases. Each test case consists of a pair of integers $[A, B]$, where $1 \leq A \leq 400{,}000$ and $A \leq B \leq 400{,}000$. The last line will contain two zeroes, to indicate the end of input and should not be processed.

Output

For each test case, output the number of reversible primes within the given range. Each case should appear on a single line by itself.

Sample Input

```
1 23
1 101
4 6
0 0
```

Sample Output

```
7
14
```