1. Consider the following algorithm to generate a sequence of numbers. Start with an integer n. If n is even, divide by 2. If n is odd, multiply by 3 and add 1. Repeat this process with the new value of n, terminating when $n = 1$. For example, the following sequence of numbers will be generated for $n = 22$:

   22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

   It is conjectured (but not yet proven) that this algorithm will terminate at $n = 1$ for every integer n. Still, the conjecture holds for all integers up to at least $1,000,000$. For an input n, the cycle-length of n is the number of numbers generated up to and including the 1. In the example above, the cycle length of 22 is 16. Given any two numbers i and j, you are to determine the maximum cycle length over all numbers between i and j, including both endpoints.

   Input

   The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than $1,000,000$ and greater than 0.

   Output

   For each pair of input integers i and j, output i, j in the same order in which they appeared in the input and then the maximum cycle length for integers between and including i and j. These three numbers should be separated by one space, with all three numbers on one line and with one line of output for each line of input.

Sample Input	Sample Output
1 10	1 10 20
100 200	100 200 125
201 210	201 210 89
900 1000	900 1000 174

2. A sequence of $n > 0$ integers is called a jolly jumper if the absolute values of the differences between successive elements take on all possible values 1 through $n - 1$. For instance,

   1 4 2 3

   is a jolly jumper, because the absolute differences are 3, 2, and 1, respectively. The definition implies that any sequence of a single integer is a jolly jumper.Write a program to determine whether each of a number of sequences is a jolly jumper.

   Input

   Each line of input contains an integer $n < 3,000$ followed by n integers representing the sequence.

   Output

   For each line of input generate a line of output saying "Jolly" or "Not jolly".

Sample Input	Sample Output
4 1 4 2 3	Jolly
5 1 4 2 -1 6	Not jolly

3. Recall the definition of the Fibonacci numbers:

$$\begin{aligned} f_1 &:= 1 \\ f_2 &:= 2 \\ f_n &:= f_{n-1} + f_{n-2} \quad (n \geq 3) \end{aligned}$$

Given two numbers a and b, calculate how many Fibonacci numbers are in the range $[a, b]$.

Input

The input contains several test cases. Each test case consists of two non-negative integer numbers a and b. Input is terminated by $a = b = 0$. Otherwise, $a \leq b \leq 10^{100}$. The numbers a and b are given with no superfluous leading zeros.

Output

For each test case output on a single line the number of Fibonacci numbers f_i with $a \leq f_i \leq b$.

Sample Input	Sample Output
10 100	5
1234567890 9876543210	4
0 0	

4. From Euclid, it is known that for any positive integers A and B there exist such integers X and Y that $AX + BY = D$, where D is the greatest common divisor of A and B. The problem is to find the corresponding X, Y, and D for a given A and B.

   Input

   The input will consist of a set of lines with the integer numbers A and B, separated with space $(A, B < 1,000,000,001)$.

   Output

   For each input line the output line should consist of three integers X, Y, and D, separated with space. If there are several such X and Y, you should output that pair for which $X \leq Y$ and $|X| + |Y|$ is minimal.

Sample Input	Sample Output
4 6	-1 1 2
17 17	0 1 17

5. Given any integer $0 \leq n \leq 10,000$ not divisible by 2 or 5, some multiple of n is a number which in decimal notation is a sequence of 1's. How many digits are in the smallest such multiple of n?

Input

A file of integers at one integer per line.

Output

Each output line gives the smallest integer $x > 0$ such that $p = \sum_{i=0}^{x} 1 \times 10^i$, where a is the corresponding input integer, $p = a \times b$, and b is an integer greater than zero.

Sample Input	Sample Output
3	3
7	6
9901	12

6. Solomon Golomb's self-describing sequence $< f(1), f(2), f(3), \cdots >$ is the only nondecreasing sequence of positive integers with the property that it contains exactly $f(k)$ occurrences of k for each k. A few moment's thought reveals that the sequence must begin as follows:

n	1	2	3	4	5	6	7	8	9	10	11	12
$f(n)$	1	2	2	3	3	4	4	4	5	5	5	6

In this problem you are expected to write a program that calculates the value of $f(n)$ given the value of n.

Input

The input may contain multiple test cases. Each test case occupies a separate line and contains an integer n ($1 \leq n \leq 2,000,000,000$). The input terminates with a test case containing a value 0 for n and this case must not be processed.

Output

For each test case in the input, output the value of $f(n)$ on a separate line.

Sample Input	Sample Output
100	21
9999	356
123456	1684
1000000000	438744
0	

7. Stan and Ollie play the game of multiplication by multiplying an integer p by one of the numbers 2 to 9. Stan always starts with $p = 1$, does his multiplication, then Ollie multiplies the number, then Stan, and so on. Before a game starts, they draw an integer $1 < n < 4,294,967,295$ and the winner is whoever reaches $p \leq n$ first.

   Input

   Each input line contains a single integer n.

   Output

   For each line of input, output one line-either

   Stan wins.

   or

   Ollie wins.

   assuming that both of them play perfectly.

Sample Input	Sample Output
162	Stan wins.
17	Ollie wins.
34012226	Stan wins.

8. A company offers personal computers for sale in N towns ($3 \leq N \leq 35$), denoted by $1, 2, \cdots, N$. There are direct routes connecting M pairs among these towns. The company decides to build servicing stations to ensure that for any town X, there will be a station located either in X or in some immediately neighboring town of X. Write a program to find the minimum number of stations the company has to build.

   Input

   The input consists of multiple problem descriptions. Every description starts with number of towns N and number of town-pairs M, separated by a space. Each of the next M lines contains a pair of integers representing connected towns, at one pair per line with each pair separated by a space. The input ends with $N = 0$ and $M = 0$.

   Output

   For each input case, print a line reporting the minimum number of servicing stations needed.

Sample Input

```
8 12
1 2
1 6
1 8
2 3
2 6
3 4
3 5
4 5
4 7
5 6
6 7
6 8
0 0
```

Sample Output

```
2
```

9. A subsequence of a given sequence S consists of S with zero or more elements deleted. Formally, a sequence $Z = z_1 z_2 \cdots z_k$ is a subsequence of $X = x_1 x_2 \cdots x_m$ if there exists a strictly increasing sequence $< i_1, i_2, \cdots, i_k >$ of indices of X such that for all $j = 1, 2, \cdots, k$, we have $x_{ij} = z_j$. For example, $Z = bcdb$ is a subsequence of $X = abcbdab$ with corresponding index sequence $< 2, 3, 5, 7 >$. Your job is to write a program that counts the number of occurrences of Z in X as a subsequence such that each has a distinct index sequence.

   Input

   The first line of the input contains an integer N indicating the number of test cases to follow. The first line of each test case contains a string X, composed entirely of lowercase alphabetic characters and having length no greater than 10,000. The second line contains another string Z having length no greater than 100 and also composed of only lowercase alphabetic characters. Be assured that neither Z nor any prefix or suffix of Z will have more than 10100 distinct occurrences in X as a subsequence.

   Output For each test case, output the number of distinct occurrences of Z in X as a subsequence. Output for each input set must be on a separate line.

   Sample Input

   ```
   2
   babgbag
   bag
   rabbbit
   rabbit
   ```

   Sample Output

   ```
   5
   3
   ```

10. Any set of n integers form $n(n-1)/2$ sums by adding every possible pair. Your task is to find the n integers given the set of sums.

Input

Each line of input contains n followed by $n(n-1)/2$ integer numbers separated by a space, where $2 < n < 10$.

Output

For each line of input, output one line containing n integers in non-descending order such that the input numbers are pairwise sums of the n numbers. If there is more than one solution, any one will do. If there is no solution, print "Impossible". . .

Sample Input	Sample Output
3 1269 1160 1663	383 777 886
3 1 1 1	Impossible
5 226 223 225 224 227 229 228 226 225 227	111 112 113 114 115
5 216 210 204 212 220 214 222 208 216 210	101 103 107 109 113
5 -1 0 -1 -2 1 0 -1 1 0 -1	-1 -1 0 0 1
5 79950 79936 79942 79962 79954 79972 79960 79968 79924 79932	39953 39971 39979 39983 39989