

# H. Ping!

Source file: **ping.{c, java, cpp}**  
Input file: **{stdin, System.in, cin}**  
Output: **{stdout, System.out, cout}**

Suppose you are tracking some satellites. Each satellite broadcasts a ‘Ping’ at a regular interval, and the intervals are unique (that is, no two satellites ping at the same interval). You need to know which satellites you can hear from your current position. The problem is that the pings cancel each other out. If an even number of satellites ping at a given time, you won’t hear anything, and if an odd number ping at a given time, it sounds like a single ping. All of the satellites ping at time **0**, and then each pings regularly at its unique interval.

Given a sequence of pings and non-pings, starting at time **0**, which satellites can you determine that you can hear from where you are? The sequence you are given may, or may not, be long enough to include all of the satellites’ ping intervals. There may be satellites that ping at time **0**, but the sequence is not long enough for you to hear their next ping. You do not have enough information to report about these satellites. Just report about the ones with an interval short enough to be in the sequence of pings.

## Input

There will be several test cases in the input. Each test case will consist of a single string on its own line, with from **2** to **1,000** characters. The first character represents time **0**, the next represents time **1**, and so on. Each character will either be a **0** or a **1**, indicating whether or not a ping can be heard at the time (**0** = No, **1** = Yes). Each input is guaranteed to have at least one satellite that can be heard. The input will end with a line with a single **0**.

## Output

For each test case, output a list of integers on a single line, indicating the intervals of the satellites that you know you can hear. Output the intervals in order from smallest to largest, with a single space between them. Output no extra spaces, and do not separate answers with blank lines.

Sample Input	Sample Output
<b>01000101101000 1001000101001000 0</b>	<b>1 2 3 6 8 10 11 13 3 6 7 12 14 15</b>