Digit Holes

Problem Statement

Tommy’s evil teacher assistant Bird has given her the following puzzle: 42 -> 1, 1337 -> 0, 669 -> 3, 1882 -> 4, 688 -> 5, 12345 -> 1, 67890 -> 5, 123 -> 0, 456 -> 2, 789 -> 3. Given this information, 45678 -> ? Thanks to her fast thinking and problem-solving intuition she has found the solution: Google. The answer turned out to be the total number of "holes" in the digits of the number's decimal representation (with no extra leading zeroes). We can see that the digits 1, 2, 3, 5, and 7 contain no holes, 0, 4, 6, and 9 each has one hole, and 8 contains two holes. Therefore the answer to the puzzle is 45678 -> 1 + 0 + 1 + 0 + 2 = 4. You want to impress Tom, so you decide to write a program that will find the correct answer for certain integers. Given an integer **number**, return the total number of holes in that number.

Input

There are multiple inputs. For each input, there is an integer representing **number(0<= number<=1000).** Input ends with EOF.

Output

For each testcase , output the total number of holes in that number

Examples

Input	Output
42 669 688 123 456 789	1 3 5 0 2 3