UGLY NUMBERS

CHALLENGE DESCRIPTION:

Credits: This challenge has appeared in a google competition before.

Once upon a time in a strange situation, people called a number ugly if it was divisible by any of the one-digit primes (2, 3, 5 or 7). Thus, 14 is ugly, but 13 is fine. 39 is ugly, but 121 is not. Note that 0 is ugly. Also note that negative numbers can also be ugly: -14 and -39 are examples of such numbers.

One day on your free time, you are gazing at a string of digits, something like:

```
123456
```

You are amused by how many possibilities there are if you are allowed to insert plus or minus signs between the digits. For example you can make:

```
1 + 234 - 5 + 6 = 236
```

which is ugly. Or

```
123 + 4 - 56 = 71
```

which is not ugly.

It is easy to count the number of different ways you can play with the digits: Between each two adjacent digits you may choose put a plus sign, a minus sign, or nothing. Therefore, if you start with D digits there are 3^(D-1) expressions you can make. Note that it is fine to have leading zeros for a number. If the string is '01023', then '01023', '0+1-02+3' and '01-023' are legal expressions.

Your task is simple: Among the 3^(D-1) expressions, count how many of them evaluate to an ugly number.

INPUT SAMPLE:

Your program should accept as its first argument a path to a filename. Each line in this file is one test case. Each test case will be a single line containing a non-empty string of decimal digits. The string in each test case will be non-empty and will contain only characters '0' through '9'. Each string is no more than 13 characters long. E.g.

```
1
9
011
12345
```

OUTPUT SAMPLE:

Print out the number of expressions that evaluate to an ugly number for each test case, each one on a new line. E.g.

```
0
1
6
64
```