2259 Do You Get the Message?

In this problem we will be using a one-bit error-correcting code called a Hamming code, which allows the receiver to correct any one-bit error in a binary sequence called a *transmission string*. The transmission string is generated by the sender from a binary sequence called the *message string*, and additional embedded bits called *parity bits*.

Let us number the transmission string bit positions from left to right starting at 1. The embedded parity bits $p_0, p_1, p_2, \dots p_i, \dots p_n$ are at positions $1, 2, 4, \dots 2i \dots 2n$ respectively, where $2n <$ length(*transmission string*). Associated with each parity bit p_i is a subset s_i of transmission string bits. Note that p_i is a member of subset s_i. Note also that these subsets are non-disjoint. The value of the parity bits are set so that the parity (sum) of all the bits in their corresponding subset is even. The implied subsets are most easily explained by giving an example: a transmission string bit at position $13 = 1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 2^3 + 2^2 + 2^0$ contributes to the parity of subsets s_3, s_2 and s_0.

For example, if the message is "`10110`", the transmission string will be laid out like

Position	1	2	3	4	5	6	7	8	9
Transmission string	p_0	p_1	1	p_2	0	1	1	p_3	0

- To make the parity of s_0 (bits at positions 1,3,5,7,9) even, p_0 is set to 0.
- To make the parity of s_1 (bits at positions 2,3,6,7) even, p_1 is set to 1.
- To make the parity of s_2 (bits at positions 4,5,6,7) even, p_2 is set to 0.
- To make the parity of s_3 (bits at positions 8,9) even, p_3 is set to 0.

Thus, the generated transmission stringcalled a Hamming codewill be 011001100, and the sender now transmits this.

We will assume that, if an error occurs during transmission, no more than one bit in the received transmission string will be in error (i.e., inverted). The receiver checks the parity of each subset. If s_i is even (i.e., correct), assign a bit b_i a value of 0, otherwise a value of 1. If the binary number corresponding to $b_n \dots b_i \dots b_2 b_1 b_0$ is zero, there were no errors in the transmission, otherwise, the binary number corresponds to the location of the error.

Going back to the example above, let us say that (unbeknown to the receiver) bit 5 was corrupted (inverted), i.e., the received transmission string is "`011011100`".

The receiver calculates the subset parities, and calculates the error position.

$b_0 = 1$ (s_o is odd)
$b_1 = 0$ (s_1 is even)
$b_2 = 1$ (s_2 is odd)
$b_3 = 0$ (s_3 is even)

The receiver discovers that the received transmission string bit at position $b_3 b_2 b_1 b_0 = 0101_2 = 5$ is in error. Thus, the corrected transmission string is "`011001100`". The (corrected) message string is "`10110`", obtained by stripping out the parity bits.

Input

The input contains a number of received binary transmission strings, one to a line, that are valid Hamming codes with no more than 1 error ($3 \leq length\ of\ string < 32,000$). The list is terminated with a transmission string of '`0`', which is not to be processed.

Output

The (corrected) binary message string is to be output, one to a line.

Sample Input

```
011011100
011001100
01101
0
```

Sample Output

```
10110
10110
11
```