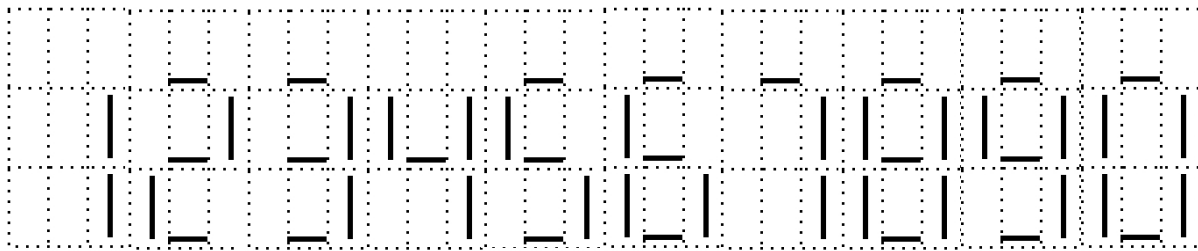


Bank Cheque Problem

Banks, always trying to increase their profit, asked their computer experts to come up with a system that can read bank cheques; this would make the processing of cheques cheaper. One of their ideas was to use optical character recognition (OCR) to recognize bank accounts printed using 7 line-segments.

Once a cheque has been scanned, some image processing software would convert the horizontal and vertical bars to ASCII bars `|' and underscores `_'.

The ASCII 7-segment versions of the ten digits look like this:



A bank account has a 9-digit account number with a checksum. For a valid account number, the following equation holds: $(d_1 + 10 * d_2 + 100 * d_3 + \dots + 10^8 * d_9) \bmod 11 = 0$. Digits are numbered from right to left like this: $d_9 d_8 d_7 d_6 d_5 d_4 d_3 d_2 d_1$.

Unfortunately, the scanner sometimes makes mistakes: some line-segments may be missing. Your task is to write a program that deduces the original number, assuming that:

- When the input represents a valid account number, it is the original number;
- At most one line is missing or extra.

For example, the following input

1 2

012345678901234567890123456

```

  _  _  _  _  _  _  _  _  _
 |  |  |  |  |  |  |  |  |
 |  |  |  |  |  |  |  |  |

```

is "123456789".

The input file contains a list of account numbers, each account number occupies 3 lines of 27 characters each (3 characters in each line for every digit).

For each test case, the output contains one line with 9 digits if the correct account number can be determined, the string ``failure" if no solutions were found and ``ambiguous" if more than one solution was found.

Sample Input

```
  | | | | | | | | |
  | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
```

Output for the above input:

ambiguous

failure

490067776