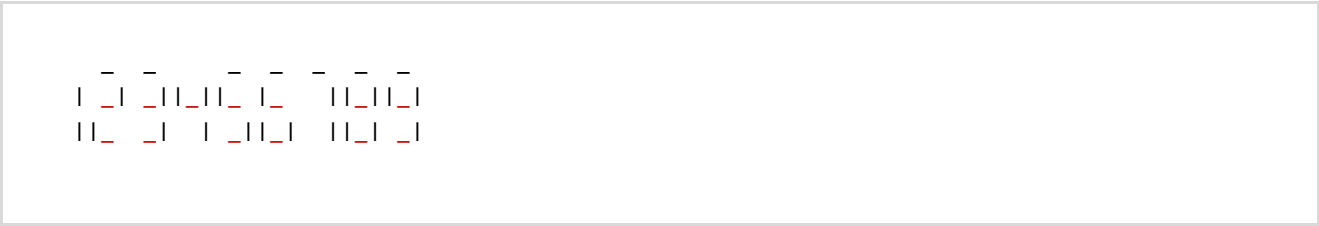


猜数字

用户故事1

假如你在一家银行工作，银行最近购买了一台智能机器，用来帮助阅读分行寄来的信件和传真。机器扫描纸质文档，并产生一个文本文件，该文本文件由如下若干条目组成：



每个条目由 4 行组成，每行有 27 个字符。每个条目前 3 行由竖线和下划线拼接为 9 位的帐号，而第四行是空白行。其中，帐号由在 0 到 9 之间的 9 个数字组成，一个普通文件包含大约 500 个条目组成。

您的第一项任务是编写一个程序，把这个文本文件解析成实际的帐号列表。例如：

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----

[illegible]

-	-	-	-	-	-	-	-	-

A diagram showing a 10x10 grid of squares. Each square is divided horizontally by a red line, creating two smaller squares. The red lines are positioned in the middle of each row, and there are 10 such lines in total, one for each row.

[illegible]

输出:

000000000
111111111
222222222
333333333
444444444
555555555
666666666
777777777
888888888
999999999
123456789

用户故事2

你很快会意识到，这个机器并不是绝对可靠，有时在扫描过程中会出现错误。接下来，需要验证您所读取的数字串是不是一个有效的帐号。有效的帐户号具有合法的校验和，计算规则如下：

账号： 3 4 5 8 8 2 8 6 5
编号： d9 d8 d7 d6 d5 d4 d3 d2 d1

计算校验和的公式为：

$$(d1+2*d2+3*d3+\dots+9*d9) \% 11 == 0$$

如上例：

$$\text{checksum} = (1*5+2*6+3*8+\dots+9*3) \% 11$$

您的第二项任务是编写代码计算给定数字串的校验和，判定其是否是有效的帐户。

用户故事3

你的老板渴望看到你的工作成果，他要求你完成一个报表，对于机器生成的每个文本文件，将其转翻译为如下的文本格式：

457508000
664371495 ERR
86110??36 ILL

在这个文件中，每行一个帐号。如果某些字符难以辨认，使用 ? 替代。并在最后一列标注标注失败原因，如果检验和失败，标注 ERR；如果存在难以辨认的数字，标注 ILL。例如：

```

- - - - -
| | | | | | | | | | |
| | | | | | | | | | |

- - - - -
| | | | | | | | | | |
| | | | | | | | | | |

- - - - -
| | | | | | | | | | |
| | | | | | | | | | |

- - - - -
| | | | | | | | | | |
| | | | | | | | | | |

```

输出:

```

000000051
49006771? ILL
1234?678? ILL
664371495 ERR

```

用户故事4

事实证明，当一个账号状态标记为 **ERR** 或 **ILL** 时，通常是因为机器硬件故障，丢失或多添加其中一个下划线或竖线，导致该数字被错误识别。举例说明，

```

- - - - -
| | | | | | | | | | |
| | | | | | | | | | |

```

如果机器漏掉一个竖线，**9** 可以是 **8**，**0** 可以是 **8**，**1** 可以是 **7**，而 **5** 可以是 **9** 或 **6**。因此，您最后一个任务是查看状态为 **ERR** 或 **ILL** 账号，并尝试通过添加或删除一个竖线或下划线来猜测它们到底应该是什么，弥补机器的错误。如果当且仅当存在一个有效的校验和，那么就毫无疑问地使用它；如果存在若干选项，它们的校验和都正确，状态标记为 **AMB**，并输出所有可能值。如果你最终还是不能弄清楚它应该是什么，最终报告 **ILL**。例如：

The diagram shows a 10x10 grid. The top row consists of 10 black horizontal line segments. The bottom row consists of 10 red horizontal line segments. Between these two rows, there are 8 vertical black line segments, each spanning the height of the grid. These vertical lines are positioned at the 2nd, 4th, 6th, 8th, and 10th columns from the left, and at the 1st, 3rd, 5th, 7th, and 9th columns from the left. This creates a pattern of black and red lines that define the structure of the grid.

A diagram consisting of two rows of ten vertical bars each. Above each bar in the top row is a short horizontal dash, except for the last bar which has a dash below it. Similarly, below each bar in the bottom row is a short horizontal dash, except for the last bar which has a dash above it.

输出:

711111111
777777177
200800000
333393333
888888888 AMB ['888886888', '888888880', '888888988']
555555555 AMB ['555655555', '559555555']
666666666 AMB ['666566666', '686666666']
999999999 AMB ['899999999', '993999999', '999959999']
490067715 AMB ['490067115', '490067719', '490867715']
123456789
000000051
490867715
1234?678? ILL