**文本文件符号识别**

# 背景

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。每个条目的前`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。

识别出文件中的符号，要求如下：

1、账号是可以校验的，具体校验方式如下：

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+...+9\*d9) % 11 == 0

上面账号中的checksum = (1\*5+2\*6+3\*8+...+9\*3) % 11，如果为0则账号正确，否则账号错误

2、输入可能会有校验和错误和字符不正确的账号，可能是机器漏掉一个竖线，如`9`可以是`8`，`0`可以是`8`，`1`可以是`7`，而`5`可以是`9`或`6`。需要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。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。

3、对于无不正确字符的账号，如果无论如何补全都计算不出正确校验和的账号，标记为ERR

4、对于有不正确字符的账号，如果无论如何补全都计算不出正确校验和的账号，标记为ILL。

以下是一个输入输出的例子：

输入是个文件，文件内如如下：

```

| | | | | | | | |

| | | | | | | | |

\_ \_ \_ \_ \_ \_ \_ \_ \_

| | | | | | | | |

| | | | | | | | |

\_ \_ \_ \_ \_ \_ \_ \_ \_

\_|| || || || || || || || |

|\_ |\_||\_||\_||\_||\_||\_||\_||\_|

\_ \_ \_ \_ \_ \_ \_ \_ \_

\_| \_| \_| \_| \_| \_| \_| \_| \_|

\_| \_| \_| \_| \_| \_| \_| \_| \_|

\_ \_ \_ \_ \_ \_ \_ \_ \_

|\_||\_||\_||\_||\_||\_||\_||\_||\_|

|\_||\_||\_||\_||\_||\_||\_||\_||\_|

\_ \_ \_ \_ \_ \_ \_ \_ \_

|\_ |\_ |\_ |\_ |\_ |\_ |\_ |\_ |\_

\_| \_| \_| \_| \_| \_| \_| \_| \_|

\_ \_ \_ \_ \_ \_ \_ \_ \_

|\_ |\_ |\_ |\_ |\_ |\_ |\_ |\_ |\_

|\_||\_||\_||\_||\_||\_||\_||\_||\_|

\_ \_ \_ \_ \_ \_ \_ \_ \_

|\_||\_||\_||\_||\_||\_||\_||\_||\_|

\_| \_| \_| \_| \_| \_| \_| \_| \_|

\_ \_ \_ \_ \_ \_ \_

|\_||\_|| || ||\_ | | ||\_

| \_||\_||\_||\_| | | | \_|

\_ \_ \_ \_ \_ \_ \_

\_| \_| \_||\_||\_ |\_ ||\_||\_|

||\_ \_| | \_||\_| ||\_| \_|

\_ \_ \_ \_ \_ \_ \_

| || || || || || || ||\_ |

|\_||\_||\_||\_||\_||\_||\_| \_| |

\_ \_ \_ \_ \_ \_ \_

|\_||\_|| ||\_||\_ | | | \_

| \_||\_||\_||\_| | | | \_|

\_ \_ \_ \_ \_ \_ \_

| \_| \_||\_| \_ |\_ ||\_||\_|

||\_ \_| | \_||\_| ||\_| \_

```

输出字符串：

```

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

```

# 设计要求

(1)使用TDD开发，进行简单清晰的表达

(2)开发语言：c/c++、java、python、scala，js

(3)提供源码、用例，另外需提供完整可执行文件包，其中入口可执行文件给定测试用txt文件，输出结果字符串，格式见上。

要求：请您以Word输出整体运作方案，并将其中要点以PPT形式进行输出，在极致挑战环节进行宣讲。