

## 号码簿

### 【问题描述】

前些年 B 君还没有手机。当时他使用一个号码簿来存储所有号码。

现在 B 君有了手机，他希望把所有号码都存入手机，但是他发现逐个号码输入的方式效率非常低，于是他找到了他的好朋友 R 君来寻求帮助。

R 君提示，如果把数字都扫描成图片，并把每个数字都处理成  $28 \times 28$  的灰度图像，则会更容易进行识别。前面的工作对 B 君来说简直小菜一碟，但是最后一步识别令 B 君一筹莫展，于是 B 君请你识别出这些数字都是什么。

B 君并不介意你看到他朋友们的电话号码，所以这是一个提交答案题。

### 【输入格式】

输入文件 *numbers.in*。

输入数据的第一行包含 1 个整数  $n$ ，表示图的数量，每幅图的行数和列数固定为 28。

接下来有  $n$  部分，每部分是一个 28 行 28 列的矩阵。矩阵的每一项是一个 0 到 255 之间的整数，表示这个位置的灰度。其中 0 表示白色，255 表示黑色。（这里和一般的灰度图片刚好相反）

### 【输出格式】

输出文件 *numbers.out* 共  $n$  行，每行包含一个数字，表示你认为这个数字是什么（数字识别的结果）。

### 【样例输入】

样例输入文件在该题目的目录中，为 *sample.in*。

### 【样例输出】

样例输出文件在该题目的目录中，为 *sample.out*。

### 【图片文件】

对于样例和测试数据的前 10000 幅图片，给出了 bmp 格式的图片，分别为 *sample.bmp* 和 *10000.bmp*。其中 *10000.bmp* 每行 100 个数字，一共 100 行。

除此之外，还有一个程序 *bmp*，你可以在与这个程序相同的目录下建立一个 *list.txt*，然后运行这个程序，你将得到 *list.bmp*。同时最多转换 100 张图，*list.txt* 格式和输入格式一样。换句话说你可以将 *numbers.in* 复制为 *list.txt* 并生成和 *sample.bmp* 同样格式的 *list.bmp*。

### 【评分方式】

如果你输出的数字个数不足 $n$ ，我们会将它与靠前的数字比对。如果个数超过 $n$ 或格式错误，你将得 0 分。

因为不同数据的重要性不同，每部分数据的分值也不同。

开始 1000 个数据，每个 1/100 分。

接下来 2000 个数据，每个 1/200 分。

接下来 3000 个数据，每个 1/300 分。

接下来 4000 个数据，每个 1/400 分。

接下来 5000 个数据，每个 1/500 分。

接下来 7000 个数据，每个 1/700 分。

接下来 8000 个数据，每个 1/800 分。

接下来 9000 个数据，每个 1/900 分。

接下来 10000 个数据，每个 1/1000 分。

接下来 11000 个数据，每个 1/1100 分。

一共 60000 个数据，总分 100 分。

因为最终成绩必须为整数，分数会四舍五入。

B 君十分宽容，并不介意你识别出错，所以出错并不会会有额外的惩罚。

### 【如何测试你的输出】

你可以使用 *checker* 测试你的输出是否符合格式要求。使用它的方法是，首先进入终端，在终端中运行下面的命令进入本题的文件夹：

```
cd numbers
```

然后运行如下的命令

```
./checker numbers.out
```

如果你的输出合法，你将看到 *checker* 成功读入了多少个整数。

如果输出不合法，你将会看到错误原因，原因包括未找到 *list.txt*， $n$  不在范围内，灰度不在范围内。

如果你希望测试其他的输出文件名，你可以将 *numbers.out* 部分换成你想要的文件名。

### 【提示】

B 君注意到手写体的数字可能有很多写法，比如 1 可以有尖，7 可能会加一横，4 右边不出头和上面封口等。我们可以在样例和前 10000 个数据中看到类似的写法。

### 【文件列表】

本题目文件比较多，每个文件的意义整理如下

文件名称	说明
<i>checker</i>	检查输出是否合法
<i>numbers.in</i>	输入文件
<i>numbers.out</i>	输出文件（需要选手自行建立）
<i>sample.in</i>	样例输入文件
<i>sample.out</i>	样例输出文件
<i>sample.bmp</i>	样例的图像
<i>10000.bmp</i>	前 10000 个数据的图像
<i>bmp</i>	可以将文本文件转换为 bmp 图像

请注意及时保存和备份，防止误操作。