

# 自动组卷系统(一)





# 1. 探索新知

## 1.1

## 文件读写

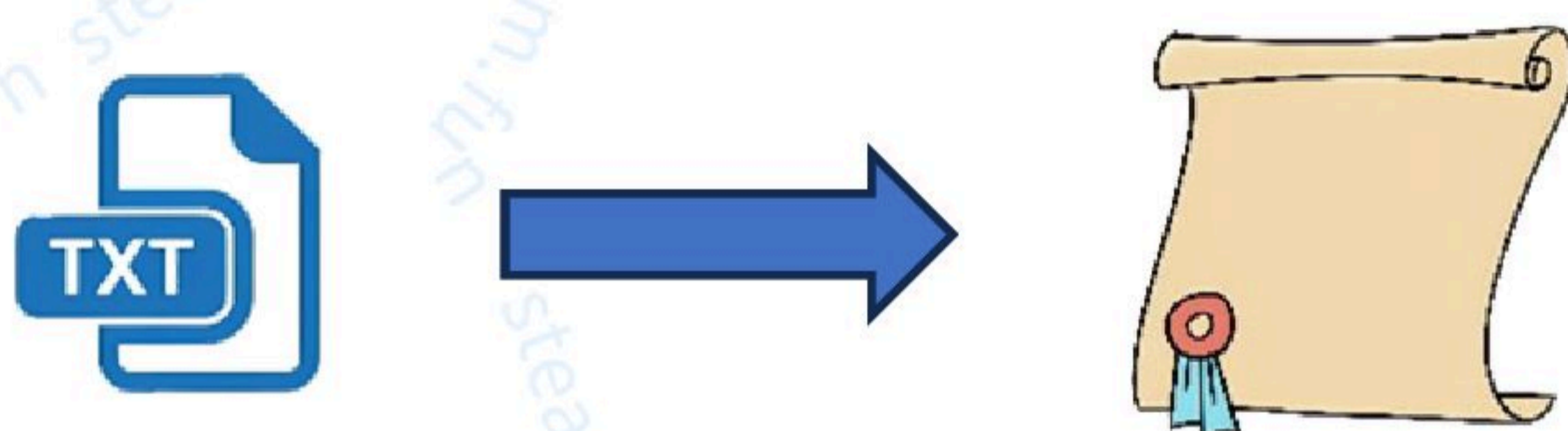
### 1. 文件是什么？

首先，我们要把文件想象成一些保存在电脑里的魔法纸张。这些魔法纸张可以保存我们的秘密，记录我们的故事，甚至可以记下我们的愿望列表。



在计算机的世界里，我们可以创建这样的纸张，也就是“文件”，然后写入一些文字，比如你写的作文或者你的画作的标题。

我们也可以随时打开这个文件，读取里面的内容，好像它是一封信或者一本书。



### 2. 如何写入文件？

当我们想要把我们的故事写入魔法纸张时，我们就需要告诉计算机“嘿，我要写点东西了！”。

我们用Python语言这样告诉它：

① 首先，我们需要打开文件。这就像是打开一本空白的记事本，准备写点什么。

【open】：  
打开

```
my_file = open('magic_paper.txt', 'w') # 'w'表示我们要写入内容
```

【file】：  
文件

## 1.1

## 文件读写

“w”：write（书写），用于写入，先清空文件再写入。

如果文件不存在，会创建一个新文件。

② 接着，我们可以写下想要保存的故事或者信息。

就像是用笔在纸上写字：

```
my_file.write('这里有一些神奇的事情发生!')
```

③ 最后，当我们完成写作后，就像合上笔记本一样，我们需要“关闭”文件：

【open】：

打开

```
my_file.close()
```

### 为什么要关闭文件？

想象一下，如果你打开很多本书，又不合上它们，那么你的房间会变得非常凌乱，并且很难找到你想要的那本书。所以，在电脑中，关闭文件是一种好习惯，可以帮助我们整理我们的电子文件，并确保我们的计算机不会弄乱它们。

### 3. 如何读取文件？

现在，如果我们想要读取刚刚保存在魔法纸张中的故事，我们就需要打开这个文件，然后将它的内容读取出来：

① 打开文件。这一次我们是为了读取内容，而不是写入内容：

```
my_file = open('magic_paper.txt', 'r') # 'r'表示我们要读取内容
```

## 1.1

### 文件读写

“r”：read（读），

就像是图书馆里的参考书，你可以阅读它，但不能在上面写字。

② 读取文件。

就像我们大声朗读书中的故事一样：

```
content = my_file.read()
```

```
print(content) # 这将会在屏幕上显示文件内容，好像它就在向我们讲述故事
```

【content】：

内容，目录

③ 关闭文件。

我们完成阅读后要合上这本魔法书：

```
my_file.close()
```

## 1.2

### 测验数据准备

"中国"："长城"，"古埃及"："金字塔"，"法国"："埃菲尔铁塔"，  
"印度"："泰姬陵"，"英国"："大本钟"，"美国"："自由女神像"，  
"意大利"："比萨斜塔"，"澳大利亚"："悉尼歌剧院"，"日本"："富士山"，  
"希腊"："帕特农神庙"，"南非"："好望角"，"丹麦"："小美人鱼铜像"，  
"巴西"："基督像"，"泰国"："大皇宫"，"加拿大"："CN塔"，  
"荷兰"："桑斯安斯风车村"，"俄罗斯"："红场"，"捷克"："布拉格城堡"，  
"沙特阿拉伯"："麦加大清真寺"，"西班牙"："圣家族大教堂"

上述国家与地标一一对应，应该用哪个数据结构存储？

【landmark】：

地标，标志

```
landmarks = {"中国": "长城", "古埃及": "金字塔", "法国": "埃菲尔铁塔",
             "印度": "泰姬陵", "英国": "大本钟", "美国": "自由女神像",
             "意大利": "比萨斜塔", "澳大利亚": "悉尼歌剧院",
             "日本": "富士山", "希腊": "帕特农神庙", "南非": "好望角",
             "丹麦": "小美人鱼铜像", "巴西": "基督像",
             "泰国": "大皇宫", "加拿大": "CN塔",
             "荷兰": "桑斯安斯风车村", "俄罗斯": "红场",
             "捷克": "布拉格城堡", "沙特阿拉伯": "麦加大清真寺",
             "西班牙": "圣家族大教堂"
            }
```

任务二新增代码如下：

# 匹配的国家 and 地标

```
landmarks = {"中国": "长城", "古埃及": "金字塔", "法国": "埃菲尔铁塔",
             "印度": "泰姬陵", "英国": "大本钟", "美国": "自由女神像",
             "意大利": "比萨斜塔", "澳大利亚": "悉尼歌剧院",
             "日本": "富士山", "希腊": "帕特农神庙", "南非": "好望角",
             "丹麦": "小美人鱼铜像", "巴西": "基督像",
             "泰国": "大皇宫", "加拿大": "CN塔",
             "荷兰": "桑斯安斯风车村", "俄罗斯": "红场",
             "捷克": "布拉格城堡", "沙特阿拉伯": "麦加大清真寺",
             "西班牙": "圣家族大教堂"
            }
```

# 1.3

## 新建文件

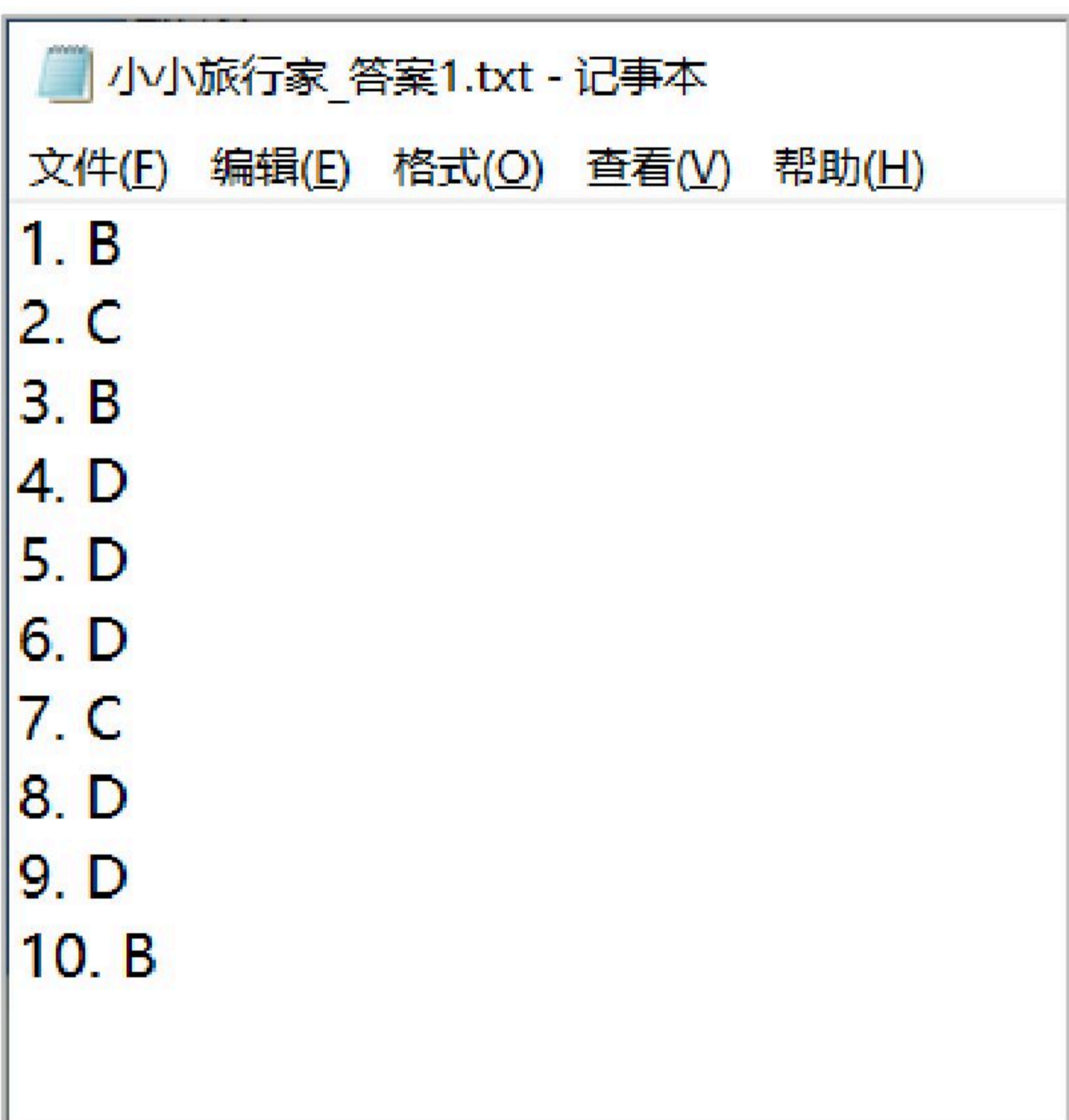
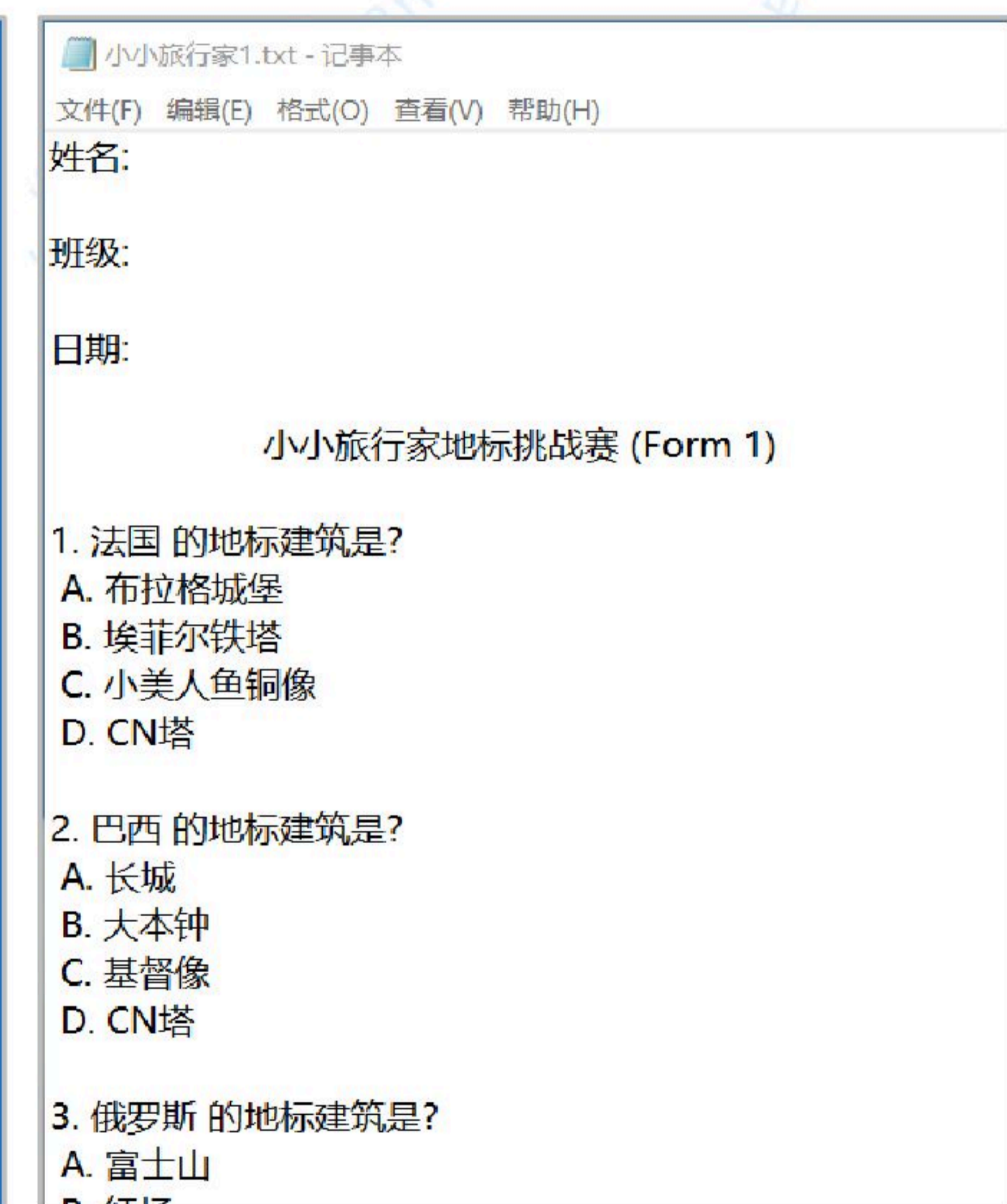
1. 假设有10个学生参加这场挑战赛，需要出10套不同的卷子

```
# 生成10套测试卷  
for i in range(10):  
    # <生成1套测试卷和答案>
```



把大问题拆分为小问题。我们先来生成1套测试卷和答案

2. 生成1套测试卷和答案



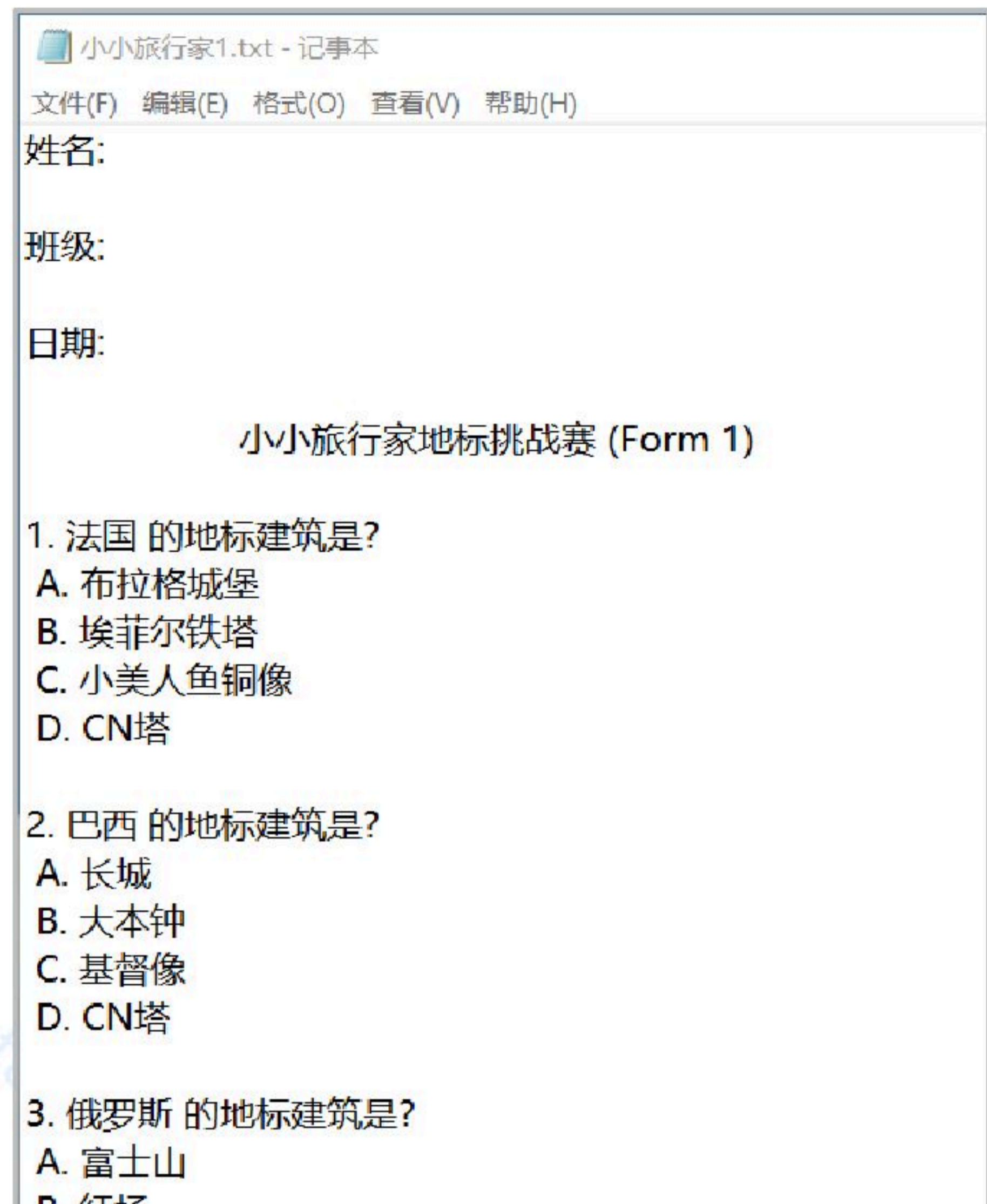
观察并总结，我们需要分哪几步来实现上述的效果？

# 1.3

## 新建文件



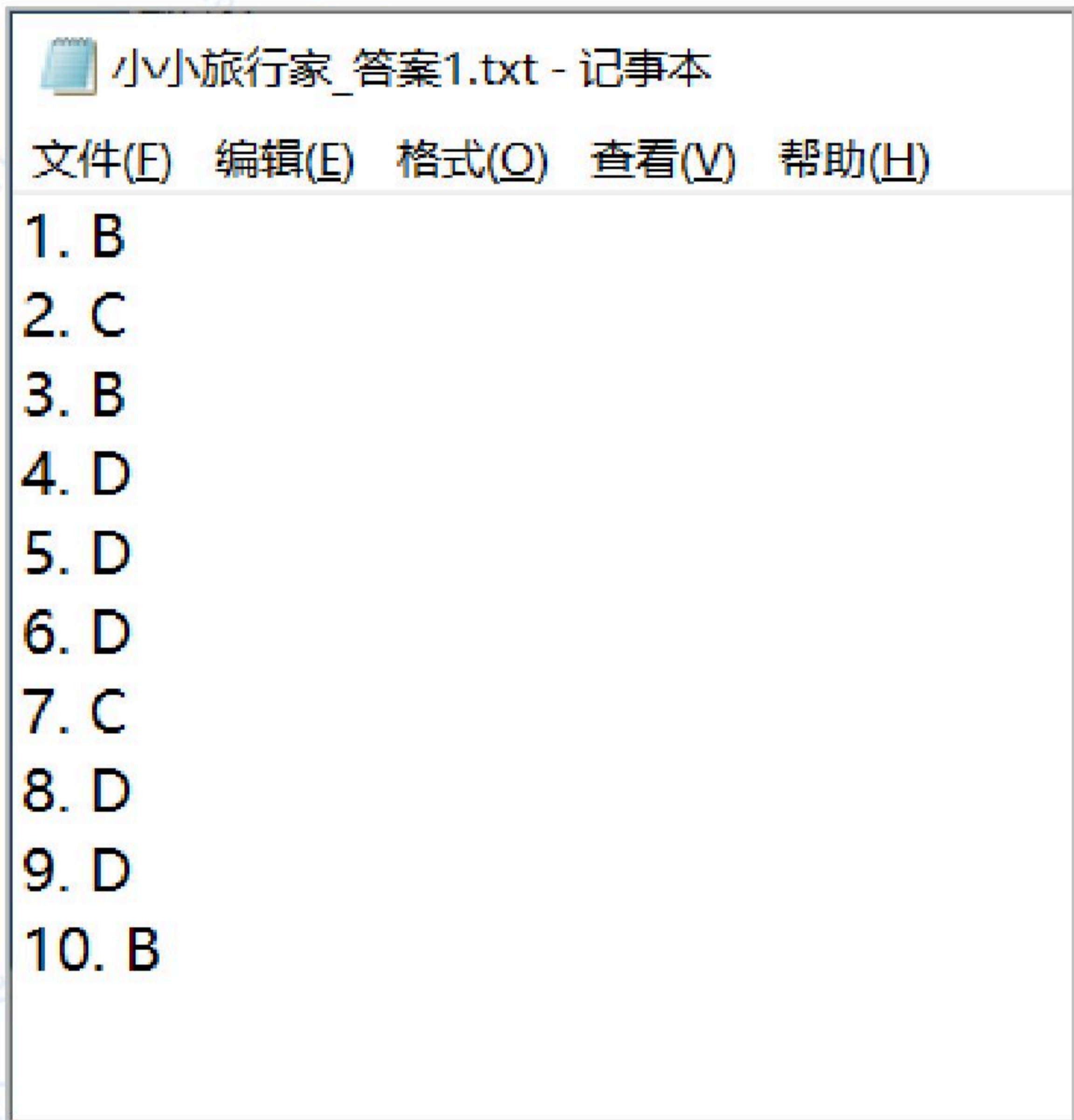
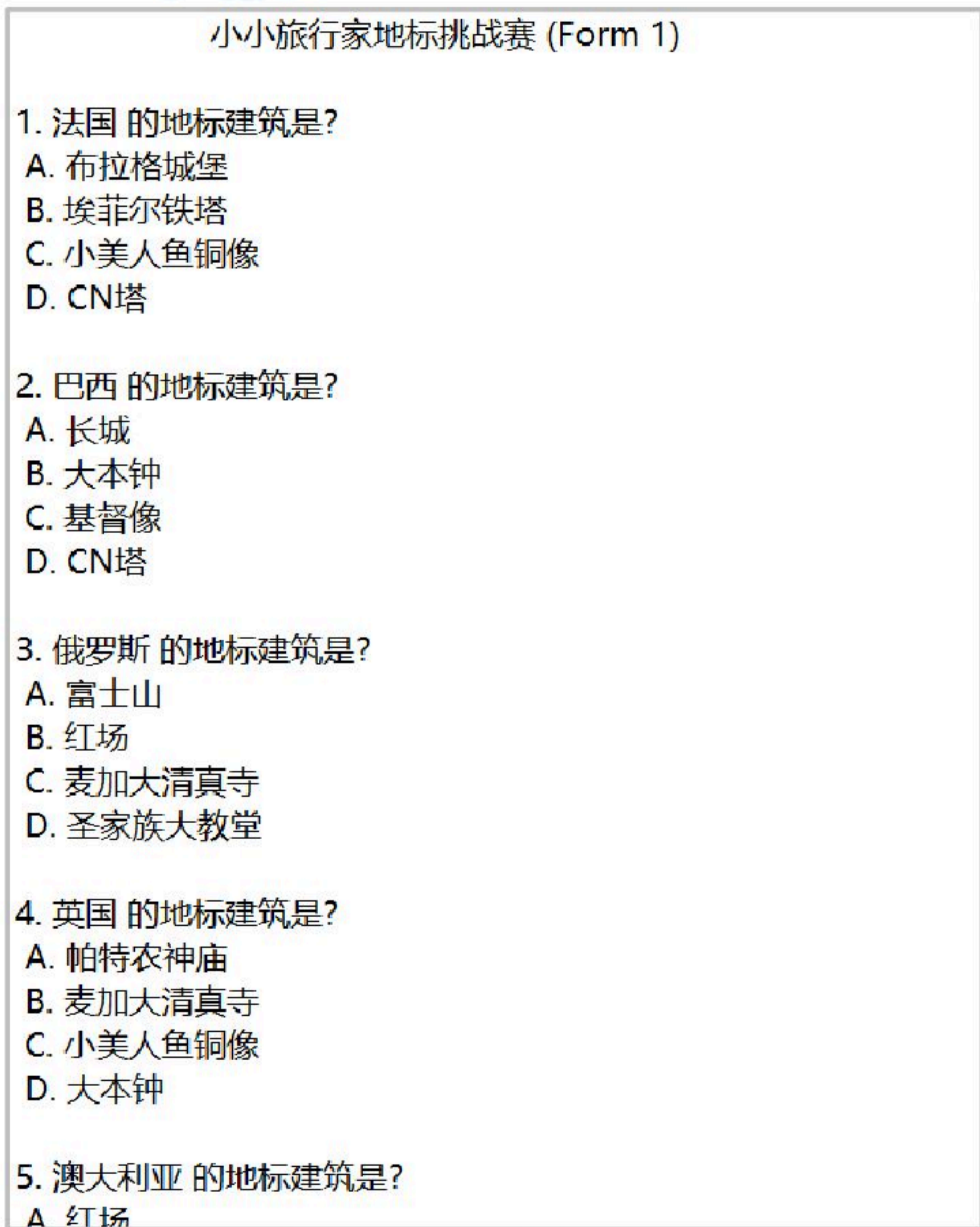
① 首先，我们需要创建一个测试卷文件以及对应答案文件



② 接着，我们需要给每套测试卷写固定格式的抬头

③ 打乱字典landmarks中国家的顺序

④ 打乱顺序后，取前10个生成题目和对应的答案，分别写入测试卷文件和答案文件



综上所述，生成1套测试卷和答案的伪代码可以这么写

# 1.3

## 新建文件

```
# 生成10套测试卷  
for i in range(10):  
    # <新建带编号的测试卷和答案文件>  
    # <写测试卷的抬头>  
    # <打乱国家的顺序>  
    # <生成10道题目和答案>
```

### 3. 新建带编号的测试卷和答案文件



测试卷的文件名为【小小旅行家N.txt】，其中 N 是测试卷的编号，从 1 开始，到 10 结束。

测试卷配套的答案文件名为【小小旅行家\_答案N.txt】，与测试卷一一对应。

还记得怎么打开文件吗？选择只读模式 'r' 还是只写模式 'w' 呢？



每次执行循环，'小小旅行家%s.txt'和'小小旅行家\_答案%s.txt'中的占位符%s都会被testNum取代，testNum是多少呢？

```
testFile = open('小小旅行家%s.txt'%(testNum), 'w')
```

```
answerFile = open('小小旅行家_答案%s.txt'%(testNum), 'w')
```

# 1.3

## 新建文件

测试卷名称	i	testNum
小小旅行家1	0	0+1
小小旅行家2	1	1+1
小小旅行家3	2	2+1
小小旅行家4	3	3+1
小小旅行家5	4	4+1
小小旅行家6	5	5+1
小小旅行家testNum	i	i+1



将testNum与i建立联系， $testNum = i + 1$ ，这行代码写在哪个位置呢？

```
# 新建带编号的测试卷和答案文件
testNum = i + 1
testFile = open('小小旅行家%s.txt'%(testNum), 'w')
answerFile=open('小小旅行家_答案%s.txt'%(testNum),'w')
```

任务三新增代码如下：

```
# 生成10套测验
for i in range(10):
    # 新建带编号的测试卷和答案文件
    testNum = i + 1
    testFile = open('小小旅行家%s.txt'%(testNum), 'w')
    answerFile = open('小小旅行家_答案%s.txt'%(testNum), 'w')
    # <待做：写测试卷的抬头>
    # <待做：打乱国家的顺序>
    # <待做：生成10道题目和答案>
```

# 1.4

## 编辑测试卷抬头



还记得如何在打开的文件中写下内容吗？  
如何留出位置，让学生填写姓名、班级和日期？

这里教大家一个新的知识——换行符 '\n'

在Python中，`\n` 是一个特殊字符，表示一个新的行。它被称为换行符。你可以在打印语句或字符串中使用 `\n` 来创建一个新的行。下面看两个例子：

```
print("Hello\nWorld")
```

```
Hello
World
```

```
print("Hello\n\nWorld")
```

```
Hello

World
```

# 1.4

## 编辑测试卷抬头



所以，如何写入这样的测试卷抬头呢？

姓名：  
班级：  
日期：

```
testFile.write('姓名:\n\n班级:\n\n日期:\n\n')
```

姓名：  
班级：  
日期：  
  
小小旅行家地标挑战赛 (Form 1)

20个空格

```
testFile.write((' ' * 20) + '小小旅行家地标挑战赛 (Form %s)' %(testNum))
```



为了美观，标题和第一题之间空了一行，该用几个换行符实现呢？自己试试吧！



```
testFile.write('\n\n')
```

## 1.4

### 编辑测试卷抬头

任务四新增代码如下：

```
# 写测试卷的抬头  
testFile.write('姓名:\n\n班级:\n\n日期:\n\n')  
testFile.write((' ' * 20) + '小小旅行家地标挑战赛 (Form %s)'%(testNum))  
testFile.write('\n\n')
```

## 完整代码

```
# 匹配的国家 and 地标
landmarks = {"中国": "长城", "古埃及": "金字塔", "法国": "埃菲尔铁塔",
             "印度": "泰姬陵", "英国": "大本钟", "美国": "自由女神像",
             "意大利": "比萨斜塔", "澳大利亚": "悉尼歌剧院",
             "希腊": "帕特农神庙", "日本": "富士山", "南非": "好望角",
             "丹麦": "小美人鱼铜像", "巴西": "基督像", "泰国": "大皇宫",
             "加拿大": "CN塔", "荷兰": "桑斯安斯风车村", "俄罗斯": "红场",
             "捷克": "布拉格城堡", "沙特阿拉伯": "麦加大清真寺",
             "西班牙": "圣家族大教堂"}

# 生成10个测验文档
for i in range(10):
    # 新建带编号的测试卷和答案文件
    testNum = i + 1
    testFile = open('小小旅行家%s.txt'%(testNum), 'w')
    answerFile = open('小小旅行家_答案%s.txt'%(testNum), 'w')
    # 写测试卷的抬头
    testFile.write('姓名:\n\n班级:\n\n日期:\n\n')
    testFile.write((' ' * 20) + '小小旅行家地标挑战赛 (Form %s)'%(testNum))
    testFile.write('\n\n')

    # <待做: 打乱国家的顺序>
    # <待做: 生成10道题目和答案>
```



## 2. 强化练习

1. 下面哪个不是中国的地标性建筑? ( )

- A. 长城
- B. 故宫
- C. 红场
- D. 东方明珠塔

2. 有关于简单文件的读写, 下列说法正确的是? ( )

- A. 可以用 Python 内置的 `open()` 函数打开文件
- B. 对文件进行读取操作, 我们可以用 `write()` 函数
- C. 对文件进行写入操作, 我们可以用 `close()` 函数
- D. 关闭文件可以用 `read()` 函数

3. 打开 `b.txt` 文件, 将 "Hello Tom!" 写入文件, 下列语句正确的是? ( )

- A. `f.write(['Hello'], [Tom!])`
- B. `f.read('Hello', ' Tom!')`
- C. `f.write('Hello Tom!')`
- D. `f.read('Hello Tom!')`

4. 下面哪个选项可以在输出中创建一个新的行? ( )

- A. `\t`
- B. `\n`
- C. `\r`
- D. `\b`

5. 根据本课代码, `testNum` 的值是什么? ( )

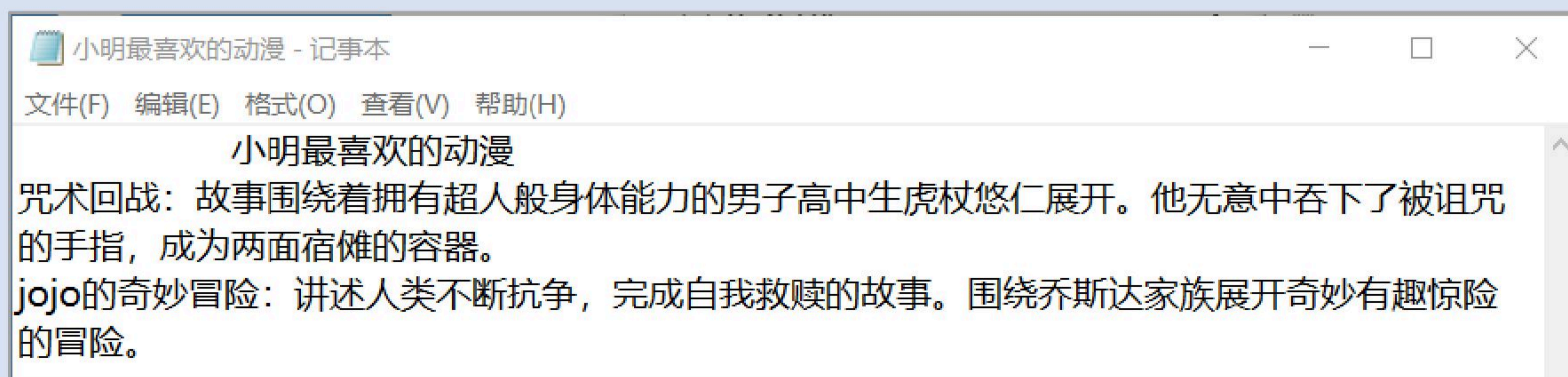
- A. `i+1`
- B. `i`
- C. `i-1`
- D. `i*2`

### 3. 术语箱

file	文件	read	读取
open	打开	landmark	地标, 标志
close	关闭	test	检验, 测试
write	书写		

### 4. 课后挑战

写一个小程序，帮助小明记录他最喜欢的动漫和对它们的描述。程序会先询问小明最喜欢的动漫数量，然后依次让小明输入每个动漫的名称和描述。最后，程序会将记录保存到一个文件中。文件示意如下。



要求如下:

1. 使用open()函数创建一个文件，并使用write()函数写入标题 "小明最喜欢的动漫"
2. 使用input()函数询问小明最喜欢的动漫数量，并将其保存在一个变量中
3. 使用一个循环，根据数量的输入设置次数，每次循环中，
  - 使用input()函数让小明输入一个动漫的名称，并将其保存在一个变量中
  - 使用input()函数让小明输入对该动漫的描述，并将其保存在一个变量中
4. 使用write()函数将动漫的名称和描述写入文件，每个动漫占用一行
5. 使用close()函数关闭文件