

准考证号：

姓名：

编程题 得分	1	2	3	4	5	6	总分	

上表由裁判在赛后填写，请参赛选手及阅卷裁判签名确认上述成绩准确无误。

选手\_\_\_\_\_、裁判\_\_\_\_\_

### 注意事项：

1. 赛场内应保持安静，参赛选手间严禁互相交谈，违者将被取消比赛资格；
2. 比赛过程中，自带的笔记本计算机及不得开启 WLAN 或蓝牙等各种无线通讯功能，违者将被取消比赛资格；
3. 比赛过程中，应将准考证及身份证件放置桌子左上角，以备监考人员随时核对、检查；
4. 参赛选手的手机在检录后必须保持关闭状态直至比赛结束，违者将被取消比赛资格；
5. 比赛过程中，笔记本电脑须调为静音；
6. 请在做题之前认真阅读题目，编程题目设置有步骤分，即只完成部分功能也可以得到相应的分数；
7. 比赛答卷时间为 120 分钟；
8. 比赛结束后，请自行整理所有自带设备及配件，并妥善带出赛场，尤其不要遗漏电源、连线等细小零部件；
9. 比赛结束时裁判现场收卷，参赛选手须核对赛卷中的准考证号和姓名准确无误；本赛卷一律交给判卷裁判，比赛选手不得自行带走，否则取消比赛成绩。

**注意：须将比赛程序集中保存到电脑上的固定目录中，命名为【准考证加名字】**

的目录下，例如：“1102030102 张明昊”，以便赛后裁判及时收集你的比赛程序。

## 一、编程题

### 第一题（难度系数 1，满分 50 分）

#### 编程实现：

用 1、3、5、8 这几个数字，能组成的互不相同且无重复数字的三位数各是多少？总共有多少个？

输入：

无

输出：

多行数字，每行一个三位数  
组成的三位数的总个数

将程序命名为“1q001”。

#### 评判标准：

- 10 分：能输出至少一个三位数字；
- 30 分：输出了多于 10 个三位数字，且确实不重复；
- 50 分：完全符合题意，即在 30 分标准的基础上，正确地输出了可组合的三位数和组合后三位数的总数量。

## 第二题（难度系数 2，满分 50 分）

### 编程实现：

打印出 1~1000 之间包含 3 的数字；

如果 3 是连在一起的（如 233）则在数字前加上&；

如果这个数字是质数则在数字后加上\*，例：（3,13\*,23\*,&33,43\*...&233\*...）。

输入：

无

输出：

按照题意输出的数字，每行一个数字；

样例输出：

```
3
13*
23*
&33
.....
```

将程序命名为“1q002”。

评判标准：

10 分：完成打印 1~1000 之间包含 3 的数字

30 分：在 10 分标准的基础上，找出其中的连 3，前面加&

50 分：在 30 分标准的基础上，找出其中的质数，后面加\*

## 第三题（难度系数 3，满分 50 分）

### 编程实现：

让用户在一次输入时输入 N 个数字（ $2 \leq N \leq 15$ ，即每次输入的数字数量不同），数字之间以“,”作为分隔。

然后组合显示：

(1)用户输入的数字个数；

(2)用户输入的最小的数字；

(3)将用户输入的数字按从大到小进行排列输出，数字之间以“,”作为分隔；

(4)如果用户输入的数字小于等于 26，则找到相对应的 26 个大写英文字母（1 对应“A”，26 对应“Z”），并拼接在一起打印显示，如果输入的数字在 1~26 之外则不显示相应字母。（例：程序输入 214，则显示输出“[bad]”）。

输入：

N 个数字， $2 \leq N \leq 15$

输出：

输入的数字个数

输入的最小数字

输入的数字从大到小排列

输入的数字所对应的字母

样例输入：

9,12,15,22,5,21,214

样例输出：

7

5

214,22,21,15,12,9,5

ILOVEU[bad]

将程序命名为“1q003”。

评判标准：

10 分：正确输出用户输入的数字个数

20 分：在 10 分标准的基础上，正确输出用户输入的最小数字

40 分：在 20 分标准的基础上，正确地将用户输入的数字按从大到小进行排序输出

50 分：完全符合题意，即在 40 分标准的基础上，正确完成显示数字对应字母的环节

#### 第四题（难度系数 3，满分 50 分）

在 Turtle 画布上画出 5 个随机大小的五角星。

具体要求：

- Turtle 画布的大小为：宽 800 像素，高 600 像素；
- 一共绘制 5 个五角星；
- 每个五角星的在画布内的显示位置是**随机**的，五角星之间允许产生重叠甚至是完全覆盖，但**每个五角星都要完整地显示在画布内**；
- 在程序运行过程中应明显看出绘制过程，建议绘图速度为 7；
- 每个五角星的线段颜色为黑色，线宽为 5，边长为[10~150]之间的随机长度，并用黄色填充；
- 提示：五角星的每个顶点的内角（锐角）为 36 度。

五角星的基本形状例如本题图-1 所示：

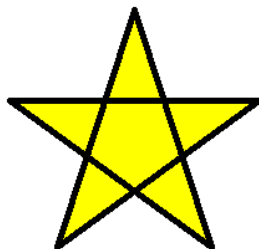


图-1

将程序命名为“1q004”。

评判标准：

10 分：至少画出了一个形状正确的五角星，不论其大小、位置、颜色和填充与否；

20 分：至少画出了一个形状、线段颜色、线宽正确的五角星；

40 分：生成了 5 个随机位置、随机大小且形状和线段颜色都正确的五角星，随机与否需多次运行来验证；

50 分：完全符合题意，即在 40 分标准的基础上，多次运行程序后，所有三角形都显示在画布范围内，且填充颜色也为黄色，也能明确看出绘画的过程。

### 第五题（难度系数 5，总分 100 分，程序 1 满分 50 分，程序 2 满分 50 分）

#### 分形树

##### 步骤 1:

利用 Turtle 画出如本题图-1 所示的分形树。

要求：

- 树木主干向上生长；
- 分形层数为 4，二叉树；
- 第一层树枝长度为 60，逐层减 6；
- 左右树枝的倾斜角度不限，最终效果与图-1 所示大致相同即可。
- 必须能看出绘图过程。

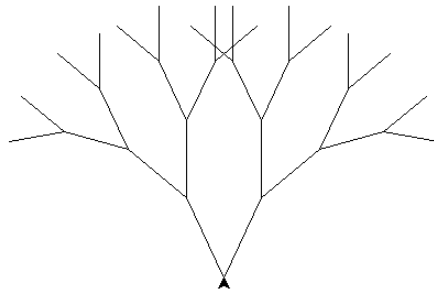


图-1

将程序命名为“1q00501”。

评判标准：

10 分：能够画出大致如图中的第一层树枝；

30 分：能够画出大致如图中的两层以上的树枝，且为二叉树；

50 分：完全符合题意，即：向上生长，二叉树，层数为 4，树枝长度每层递减，能看出绘图过程。

##### 步骤 2:

一、修改步骤 1 中生成分形树的方法：

- 分型层数改为 4~6 之间的随机数；
- 第一层树枝长改为 30~60 之间的随机数；

二、绘制分形树丛

- 在 800X600 的屏幕中绘制分形树丛；
- 树木数量为 50 棵；
- 每棵分形树的树根位置为屏幕范围内随机坐标点，树枝允许伸出屏幕窗口外；
- 以屏幕中心垂直线为分界，位于屏幕左侧的树木向左倾斜，位于屏幕右侧的树木向右倾斜；
- 绘制过程瞬间完成（即看不到绘画过程）。

整体绘制样例如本题图-2 所示。

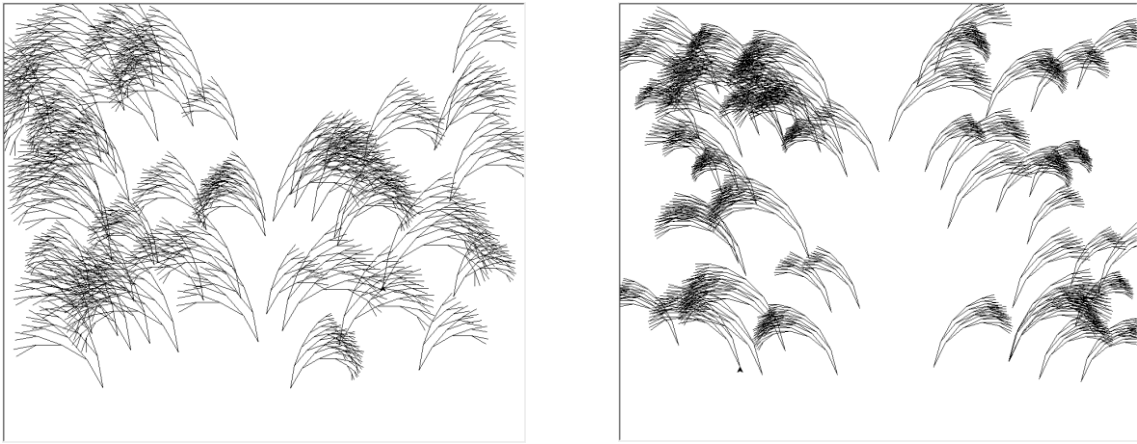


图-2 分形树丛绘制样例

将程序命名为“1q00502”。

评判标准：

- 0 分：步骤 1 没有得到满分；
- 10 分：能在画布中画出 50 颗随机位置的树木，即使朝向相同、层数固定；
- 30 分：在 10 分标准的基础上，能画出层数随机、树枝长短随机的分形树；
- 50 分：完全符合题意，即在 30 分标准的基础上，能画出符合题意要求朝向倾斜的分形树，且绘画过程瞬间完成（即看不到绘画过程）。

## 第六题（难度系数 6，满分 120 分）

编程实现：商品管理系统

预先准备：

建立 `userpass.txt`

预先建立并存储用户信息文件。文件格式：以“,”分隔的一组用户名和密码信息。例如：

`zhangsan,123456`

建立 `goods.txt`

以自定义格式存储 3 个商品的信息（每个商品包含：商品编号，商品名称，商品类型，商品库存）。

文件内容样例（仅供参考）：

`001,商品 1,类型 1,10;002,商品 2,类型 1,5;003,商品 3,类型 2,20`

系统登录：

- 显示一个 6 位随机数（验证码），并提示用户输入用户名、密码和刚刚显示的 6 位验证码；
- 如果用户输入的用户名和密码与 `userpass.txt` 文件中的信息相符合，且验证码也和提示的 6 位验证码相符合，则继续程序进入系统主界面的初始状态（见下文中的“系统主界面”->“初始状态”部分），例如以下过程：

说明：以  为背景的是内容程序输出， 内容为用户输入：

您的登录验证码为 `386597`

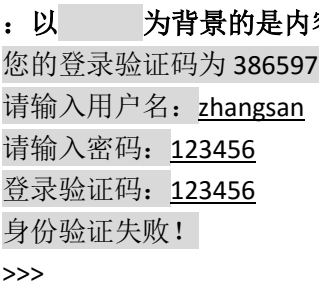
请输入用户名：`zhangsan`

请输入密码：`123456`

登录验证码：`386597`

身份验证通过，欢迎登录！

- 如果用户输入的用户名和密码与 userpass.txt 文件中的信息不符合，或者验证码和提示的 6 位验证码不符合，则登陆不成功，提示身份验证失败，退出程序。例如以下过程：

说明：以  为背景的是内容程序输出，\_\_\_\_ 内容为用户输入：

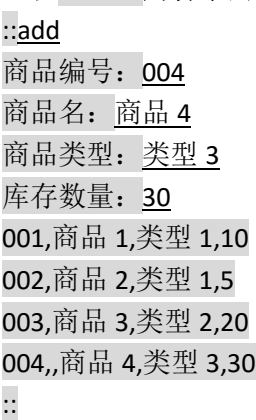
```
您的登录验证码为 386597
请输入用户名: zhangsan
请输入密码: 123456
登录验证码: 123456
身份验证失败!
>>>
```

系统主界面：

- **初始状态：**显示从文件中读取所有商品的信息（包含商品编号，商品名称,商品类型，库存数量），并以“::”提示输入命令。
- 如果输入“add”，则添加商品。要求客户输入商品编号，商品名称，商品类型，库存数量。输入时须判断所有输入项不能为空，且库存数量必须为数字格式，输入后将新添加的商品存入文件，并回到**初始状态**。
- 如果输入“count”，则计算并输出所有商品的库存数量之和。并回到**初始状态**。
- 如此循环。

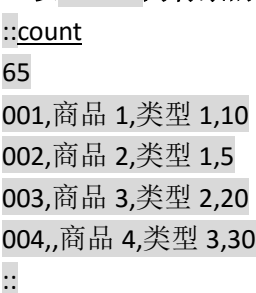
注意：在程序运行过程中，不论用户如何输入都不会出现红色 error 导致的程序异常结束（利用 try/except 语句）。

**add 命令**的输入输出过程可自行设计，输入输出样例（仅供参考）例如：

说明：以  为背景的是内容程序输出，\_\_\_\_ 内容为用户输入：

```
::add
商品编号: 004
商品名: 商品 4
商品类型: 类型 3
库存数量: 30
001,商品 1,类型 1,10
002,商品 2,类型 1,5
003,商品 3,类型 2,20
004,,商品 4,类型 3,30
::
```

**count 命令**的执行过程例如：

说明：以  为背景的是内容程序输出，\_\_\_\_ 内容为用户输入：

```
::count
65
001,商品 1,类型 1,10
002,商品 2,类型 1,5
003,商品 3,类型 2,20
004,,商品 4,类型 3,30
::
```

将程序命名为“1q006”。

**评判标准：**

20 分：实现从 userpass.txt 中读取用户信息并验证用户输入的用户名/密码的有效性；通过修改 userpass.txt 文件内容来验证；

30 分：在 20 分标准的基础上，实现了显示 6 位随机数（验证码）并验证用户输入的验证码是否与之相符的功能；

50 分：在 30 分标准的基础上，实现从文件中读取 3 个商品信息并显示在 Python 控制台中的功能；

70 分：在 50 分标准的基础上，实现添加商品（add 命令）写入文件，验证非空及数据类型，并再次从文件中读取所有商品显示的功能，回到系统主界面的初始状态，且 add 命令可多次执行；

100 分：在 70 分标准的基础上，实现商品总库存量计算（count 命令），并显示文件中所有商品，重新回到系统主界面的初始状态；且 count 命令可多次执行；

120 分：在 100 分标准的基础上，整个程序运行过程中，不论用户如何输入都不会出现红色 error 错误导致的程序出错退出。

**再次强调：比赛结束后，本赛卷一律交给现场裁判，参赛选手不得带走，否则取消比赛成绩。**