实验二 PYTHON 数据结构与流程控制

# 一、目的和要求

1. 熟悉 Python 的流程控制；
2. 熟悉 Python 的数据结构；
3. 掌握 Python 语言基本语法；

# 二、实验环境

1. 操作系统不限；
2. Python IDLE、PyCharm 等开发环境不限。

# 三、实验内容

**（一）验证实验（每个同学完成，不写入报告）**

1. Python 流程控制
2. 运行调试第三章课后练习的程序设计题，检查运行结果是否正确，记录实验结果。
3. 生成含有 20 个 1 到 100 之间随机数的、所有元素不相同的列表。
4. 安装 Swampy 扩展库，阅读和运行 koch.py，理解 Koch 曲线绘制的原理。
5. Python 序列对象
6. 运行调试第二章各小节例示代码及课后练习的程序设计题，检查运行结果是否正确，记录实验结果。
7. 阅读和运行 reversepair.py：（a）比较该代码和你的小组在实验二中所完成的反序词代码；（b）阅读和自学 Python 文档中模块 bisect 的使用方法。
8. 用嵌套循环编写 Python 源码生成 100 以内的所有素数，并且将这段源码修改为一条列表推导式；
9. 运行下面代码，自学 Counter 对象的使用

from collections import Counter cnt = Counter()

for word in ['red', 'blue', 'red', 'green', 'blue', 'blue']:

cnt[word] += 1

cnt

1. Python 文件读写：参照课本第 7 章各小节，熟悉 Python 的文件读写方法。

# （二）设计实验（小组验收，算法设计和测试写入实验报告）

1. Python 流程控制：编写循环控制代码用下面公式逼近圆周率(精确到小数点后 15 位)，并且和 math.pi 的值做比较

。

1. Python 流程控制：阅读 https://en.wikipedia.org/wiki/Koch\_snowflake，通过修改 koch.py 绘制其中一种泛化的 Koch 曲线。
2. 生日相同情形的概率分析：
3. 生成 M（M>=1000）个班级，每个班级有 N 名同学，用 input 接收 M

和 N；

1. 用 random 模块中的 randint 生成随机数作为 N 名同学的生日；
2. 计算 M 个班级中存在相同生日情况的班级数 Q，用 P=Q/M 作为对相同生日概率的估计；
3. 分析 M，N 和 P 之间的关系。
4. 参照验证实验1 中反序词实现的例示代码，设计Python 程序找出words.txt 中最长的“可缩减单词”（所谓“可缩减单词”是指：每次删除单词的一个字母，剩下的字母依序排列仍然是一个单词，直至单字母单词’a’或者‘i'）。

提示：

1. 可缩减单词例示：

sprite —〉spite —〉spit—〉pit—〉it—〉i

1. 如果递归求解，可以引入单词空字符串’’作为基准。
2. 一个单词的子单词不是可缩减的单词，则该单词也不是可缩减单词。因此，记录已经查找到的可缩减单词可以提速整个问题的求解。