**第五周任务**

本周主要是List的各项操作。

1. 首先，请务必熟练掌握ppt和对应ipynb文件里面的各项命令和代码示例
2. 清楚的了解语法的使用条件和可能要规避的bug（比如index error, value error，循环的执行规则，循环+list不能修改待遍历的list之类的，等等）
3. 做到能通过代码自己写出程序运行结果
4. 能自己复现代码示例

**分别**用for, while, range, list这4种语法完成以下练习，熟练掌握基本语法的使用。**不需要提交，自己练习。**

1. 实现函数def my\_sm(n): 返回1+2+…+n的值，不准用公式 （1+n）\*n/2
2. 实现函数def my\_AG(a, d, n): 返回数列的和，不准用公式
3. 实现函数 def my\_power(x, n): 返回 的值，n为整数。不准用x\*\*n
4. 实现函数 def my\_polynomial(a, x, n): 其中a是一个列表，求多项式

的值

1. 将一个10进制整数x转换为任意基数b的b进制整数 def dec\_convert(x, b)
2. 计算n!. def factorial(n)

**List问题，不需要用四种语法实现，包装成一个函数，需要提交**

1. 给定一个list的数，求其所有数的 和 与 乘积（空列表和为零，乘积为1）
2. 给定两个列表，判断两个列表中是否有一个包含另一个的所有元素，返回True以及较大的list, 否则返回False. （可以考虑直接用in来做，也可以考虑更复杂的sort方法）

def compare\_list(lst1, lst2): # return list1 or list2, if list1==list2, return list1

1. Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number.

The numbers obtained should be printed in a comma-separated sequence on a single line. （可以用print里面的sep选项指定分隔符，并且不用考虑print的长度）

1. With a given list (such as [12, 24, 35, 24, 88, 120, 155, 88, 120, 155]), write a program to print this list after removing all duplicate values with original order reserved. （可以遍历原列表并且开一个新列表不断append不重复的元素，并且可以考虑用sort对原列表进行预处理。）
2. 用列表实现如下两个函数

给定一个 正整数，将整数中的数字进行反转；

给定一个 负整数，将整数中的数字进行反转。

1. 请实现以下功能：写出一个函数，该函数的返回值为两个lists的交集，如果没有交集，则返回False并打印出“no intersection” （可以用in死做，或者用排序sort并且同时遍历两个排好序的列表）

~~提示：split()~~

输入：[123, 456]

[345, 456]

输出：[456]

1. Write a Python program to construct the following pattern, using a nested for loop.

\*   
\* \*   
\* \* \*   
\* \* \* \*   
\* \* \* \* \*   
\* \* \* \*   
\* \* \*   
\* \*   
\*

def print\_star(n): # n =5 above （可以考虑用不止一个循环实现，这题可以不用列表）

1. Given a positive integer n, count the total number of digits in a number. For example, n = 12333219, then 1:2, 2:2, 3:3, 9:1

def count\_digits(n): # return a list with len()=10

1. Find the sum of the series 2 +22 + 222 + 2222 + ... + 2…..2: there are n 2’s in the last term.

def sum\_of\_two(n):

1. 使用list创建两个矩阵如下，即1，2，…,10的排列

输出这两个矩阵（其中A的每一行都行向量(1, 2, ..., 10)， B的每一列都是 (1,2,...,10)转置）

1. 给定一个列表A，假定A具有结构：存在0<=i<=n-1(n为列表长度)，满足A[0]<=A[1]<= ... <= A[i] >= A[i+1]>=...>=A[n-1], 输入A，输出满足上述条件的最小的i

def find\_min(lst):

1. Sieve of Eratosthenes (筛法)

<https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes>

1. Write a function to determine whether a given integer is prime or not.
2. Write a function to find all the primes less than 1,000,000

Solution: open a list up to 1,000,000.

## Linux学习

Linux一个最便捷的地方在于管理系统上面的软件包。以前Linux系统提供了apt-get 命令来管理软件包，后来经过优化，提供了新的命令：apt (Advanced Packaging Tool).

譬如, 我们如果需要安装vim编辑器，我们仅仅需要一条命令

sudo apt install vim

可以对比下，windows下面需要多少道流程。

同理，卸载vim

sudo apt remove vim

apt的命令格式为：

* + - 1. 列出所有可更新的软件清单命令：sudo apt update
      2. 升级软件包：sudo apt upgrade
      3. 列出可更新的软件包及版本信息：apt list --upgradeable
      4. 升级软件包，升级前先删除需要更新软件包：sudo apt full-upgrade
      5. 安装指定的软件命令：sudo apt install <package\_name>
      6. 安装多个软件包：sudo apt install <package\_1> <package\_2> <package\_3>
      7. 更新指定的软件命令：sudo apt update <package\_name>
      8. 显示软件包具体信息,例如：版本号，安装大小，依赖关系等等：sudo apt show <package\_name>
      9. 删除软件包命令：sudo apt remove <package\_name>
      10. 清理不再使用的依赖和库文件: sudo apt autoremove
      11. 移除软件包及配置文件: sudo apt purge <package\_name>
      12. 查找软件包命令： sudo apt search <keyword>
      13. 列出所有已安装的包：apt list --installed
      14. 列出所有已安装的包的版本信息：apt list --all-versions

在实际使用中，我们不用去记住这些命令，只需要记住apt就够了。经常使用的有apt update, apt upgrade, apt remove, apt purge, 这些命令的作用，顾名思义。在使用中，如果遇到不会的命令，直接搜索(例如https://www.runoob.com/linux/linux-tutorial.html)，以后也会介绍Linux的帮助系统。

**安装vim**

输入vim，发现没有安装

A screenshot of a computer screen

Description automatically generated

输入命令 sudo apt install vim 安装vim，在选项中输入 Y， 代表yes。然后apt就会自动安装vim。 （在安装前，可以先使用sudo apt update检查下更新）

A screenshot of a computer

Description automatically generated

安装完成后就可以运行vim。

A screenshot of a computer

Description automatically generated

VIM和EMACS是过去最流行的两款editor之一，现在可能vs code更讨喜，毕竟vim和emacs都是40多年前开发的，虽然不断推陈出新，但是不如vs code从头设计来的好。当然了，vim和emacs我们至少要学一种，本门课程的后续章节会简要介绍vim的使用。

如何退出vim？ 先输入：, 然后输入q。记住不要暴力关闭shell。

如何安装jupyter lab，windows上面的安装经历可能让大家都心有余悸。

sudo apt install jupyter

安装完成后，输入 jupyter notebook就可以启动了

ipython是传统python 控制台解释器的加强版，提供了更强大的的功能。

sudo apt install ipython3

通过ipython3启动，可以使用更友好的界面

A computer screen with white text

Description automatically generated

Linux安装包非常便利，这也是windows向Linux靠拢，开发WSL的原因。2-3年内，windows应该也可以做到用apt安装Linux包。

卸载软件，以vim为例

sudo apt remove vim

A screenshot of a computer program

Description automatically generated

卸载完后，可以运行sudo apt autoremove清理不需要的软件包