**If逻辑判断**

1.1

编写程序计算购买图书的总金额：用户输入图书的定价和购买图书的数量，并分别保存到一个 float 和一个 int 类型的变量中，然后根据用户输入的定价和购买图书的数量，计算购书的总金额并输出。其中，图书销售策略为：正常情况下按 9 折出售，购书数量超过 10本打 8.5 折，超过 100 本打 8 折

**num = int(input('请输入购买书的数量:')) # 书的数量  
price = float(input('请输入购买书的价格:')) # 书的价格  
if 10 < num <= 100:  
 print((num \* price) \* 0.85)  
elif num > 100:  
 print((num \* price) \* 0.8)  
else:  
 print((num \* price) \* 0.9)**

1.2

动物园想在新建一个三角形的人工湖，一是为了养鱼美观，二是可以循环水资源。从键盘输入三条边 A、B、C 的边长，请编程判断能否组成一个三角形。（可以组成三角形的要素，两边之和大于第三边，两边之差小于第三边）

要求：如果三条边长 A、B、C 能组成三角形的话，输出 YES，否则 NO。

**a = float(input('请输入A边:'))  
b = float(input('请输入B边:'))  
c = float(input('请输入C边:'))  
if a + b > c and a + c > b and b + c > a:  
 print('YES')  
else:  
 print('NO')**

1.3

商店A 准备在今年夏天开始出售西瓜，西瓜的售价如下，20 斤以上的每斤 0.85 元；重于 15 斤轻于等于 20 斤的，每斤 0.90 元；重于 10 斤轻于等于 15 斤的，每斤 0.95 元；重于 5 斤轻于等于 10 斤的，每斤 1.00 元；轻于或等于 5 斤的，每斤 1.05 元。现在为了知道商店是否会盈利要求 A 公司帮忙设计一个输入西瓜的重量和顾客所付钱数，输出应付货款和应找钱数的程序。

**weight = float(input('请输入西瓜的重量:'))  
price = float(input('顾客所付钱数：'))  
repay = 0  
money = 0  
if weight > 20:  
 money = weight \* 0.85  
elif 15 < weight <= 20:  
 money = weight \* 0.9  
elif 10 < weight <= 15:  
 money = weight \* 0.95  
elif 5 < weight <= 10:  
 money = weight \* 1  
else:  
 money = weight \* 1.5  
repay = price - money  
if repay < 0:  
 print(f'商品的价格是{money}元')  
 print(f'你所支付的钱不够，你还需支付{-repay}元')  
elif repay == 0:  
 print('你所支付的钱刚刚好')  
else:  
 print(f'商品的价格是{money}元')  
 print(f'找零{repay}元')**

1.4

对学生成绩进行统计和数据分析可以发现学生对知识的掌握情况，以便教师根据分析的结果调整教学内容和重难点，现在需要完成以下任务来实现成绩分析系统。

输入一个百分制的成绩 t，将其转换成对应的等级然后输出，具体转换规则如下：

90~100 为 A

80~89 为 B

70~79 为 C

60~69 为 D

0~59 为 E

要求：如果输入数据不在 0~100 范围内，请输出一行错误提示：“数据超出范围！”

**grade = float(input('请输入成绩:'))  
if 90 <= grade < 100:  
 print('A')  
elif 80 <= grade < 90:  
 print('B')  
elif 70 <= grade < 80:  
 print('C')  
elif 60 <= grade < 70:  
 print('D')  
else:  
 print('E')**

1.5

从键盘接收一个十一位的数字，判断其是否为尾号 5 连（最后5个数一样）的手机号。规则：第 1 位是 1，

第二位可以是数字 358 其中之一，后面 4 位任意数字，最后 5 位为任意相同的数字。例如：

18601088888、13912366666 则满足。

注意：不满足的输出“false”，满足要求的输出“true”。

**phone = int(input('请输入手机号码:'))  
one = phone // 10000000000 # 第一位  
two = phone // 1000000000 % 10 # 第二位  
last\_five = [phone % 10, phone % 100 // 10, phone % 1000 // 100, phone % 10000 // 1000,  
 phone % 100000 // 10000] # 最后五位  
if one == 1: # 第 1 位是 1  
 if two == 3 or two == 5 or two == 8: # 第二位可以是数字 358 其中之一  
 if sum(last\_five) == last\_five[0] \* 5: # 最后5位为任意相同的数字  
 print("true")  
 else:  
 print("false")  
 else:  
 print("false")  
else:  
 print("false")**

1.6

按顺序输入正方形的边长（a），长方形的长（l）和宽（d），以及圆的半径（r），计算并比较它们哪个图形面积更大，输出面积最大的图形。

例如：输入 1 3 4 1，输出：长方形

**a, l, d, r = (input('按顺序输入正方形的边长（a），长方形的长（l）和宽（d），以及圆的半径（r）:')).split(' ')  
a = float(a)  
l = float(l)  
d = float(d)  
r = float(r)  
area = [a \* a, l \* d, r \*\* 2 \* 3.14]  
graphics = ['正方形', '长方形', '圆形']  
print(graphics[area.index(max(area))])**

1.7

整除判断游戏能显著提高小朋友的逻辑思维能力，问题要求如下：

•能同时被 3、5、7 整除

•能同时被 3、5 整除

•能同时被 3、7 整除

•能同时被 5、7 整除

•只能被 3、5、7 中的一个整除

•不能被 3、5、7 任一个整除

输入一个整数，输出满足对应条件的结果。要求：使用分支结构语句实现。

**data = int(input('请输入一个整数：'))  
if data % 3 == 0 and data % 5 == 0 and data % 7 == 0:  
 print('能同时被 3、5、7 整除')  
elif data % 3 == 0 and data % 5 == 0:  
 print('能同时被 3、5 整除')  
elif data % 3 == 0 and data % 7 == 0:  
 print('能同时被 3、7 整除')  
elif data % 5 == 0 and data % 7 == 0:  
 print('能同时被 5、7 整除')  
elif data % 5 == 0 or data % 7 == 0 or data % 3 == 0:  
 print('只能被 3、5、7 中的一个整除')  
else:  
 print('不能被 3、5、7 任一个整除')**

1.8

判断一个整数是否为“水仙花数”。所谓“水仙花数”是指一个三位的整数，其每个单独数字的立方和等于该数本身。例如：153 是一个“水仙花数”， 例如 153=1\*\*3＋5\*\*3＋3\*\*3。

如果是水仙花数就输出YES,否则就输出No

**data = int(input('请输入一个三位的整数：'))  
if data == (data // 100) \*\* 3 + (data // 10 % 10) \*\* 3 + (data % 10) \*\* 3:  
 print('YES')  
else:  
 print('NO')**

**循环 （包含for while 列表 字典知识）**

注意：循环一般和列表字典有联系，请先掌握此方面知识

1.1

从键盘接受一个正整数，列出该数字的中文表示格式，例如：键盘输入 123，打印出一二三；键盘输入 3103，打印出三一零三.（考验循环和列表的索引使用）

**data = int(input('请输入一个正整数：'))  
cn\_num = ['零', '一', '二', '三', '四', '五', '六', '七', '八', '九']  
print\_answer = []  
while data:  
 print\_answer.append(cn\_num[data % 10])  
 data //= 10  
for i in range(len(print\_answer) - 1, -1, -1):  
 print(print\_answer[i], end='')**

1.2

门票的序列号必定是系统里总序列的子序列，请你核对门票的真实性。

从键盘接收两个字符串a 和b，请你判断字符串a 是否包含字符串b，是的话输出“Yes”，否则输出“No”。有多组测试用例，每个测试用例占一行，两个字符串之间用空格隔开。

例如：输入JavaStudy Java 则输出Yes Student School 则输出 No

注意 ：判断后者是否存在与前者里面 请用循环完成

**a, b = input('按顺序输出字符串a,b:').split(' ')  
for i in range(len(a) - len(b) + 1):  
 if b == a[i:i + len(b)]:  
 print('YES')  
 break  
else:  
 print('NO')**

1.3 中国结

公司现在需要打印中国结的主结(位于中间，最大的那一个结)，为了打印出漂亮新颖的主结，于是设计打印主结的长度满足可以被 7 整除这个条件。现在公司需要统计某个范围内

能被 7 整除的整数的个数，以及这些能被 7 整除的数的和。

从键盘上输入一个整数 N，输出 1~N 之间能被 7 整除的整数的数，以及这些能被 7 整除的数的和

**sum\_seven = 0  
data = int(input('请输入一个正整数N：'))  
for i in range(1, data):  
 if i % 7 == 0:  
 sum\_seven += i  
 print(i, end=' ')  
print(f'\n被 7 整除的数的和:{sum\_seven}')**

1.4 项列求和

有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13 … 求出这个数列的前 20 项之和。要求：利用循环计算该数列的和。注意分子分母的变化规律。

注意：

a1=2, b1=1, c1=a1/b1; a2=a1+b1, b2=a1, c2=a2/b2; a3=a2+b2, b3=a2, c3=a3/b3;

…

s = c1+c2+…+c20;

s 即为分数序列：2/1，3/2，5/3，8/5，13/8，21/13 … 的前 20 项之和。

**num\_mom = 2 # 分母  
num\_sun = 1 # 分子  
num\_sum = 0 # 序列之和  
temp = 0 # 替换值  
for i in range(20):  
 num\_sum += num\_sun / num\_mom  
 temp = num\_mom  
 num\_mom = num\_sun  
 num\_sun = num\_sun + temp  
print(num\_sum)**

1.5

一个球从 100 米高度自由落下，每次落地后反弹回原高度的一半，再落下，再反弹。求它在第十次落地时，球共经过多少米? 第十次反弹多高?

**sum\_h = 0  
h = 100  
for i in range(10):  
 sum\_h += h#相加下落的高度  
 h = h / 2  
 sum\_h += h#相加反弹的高度  
print(f'球经过{sum\_h-h}米')  
print(f'第十次反弹{h}米高')**

1.6

求奇数、

编写程序实现：从键盘输入正整数 s，从低位开始取出 s 中的奇数位上的数，依次构成一个新数 t，高位仍放在高位，低位仍放在低位，最后在屏幕上输出 t。例如，当 s 中的数为 7654321 时，t 中的数为 7531。

**num = int(input('输入一个正整数:'))  
i = 1 # 循环次数判断奇、偶  
remainder = 0 # 余数  
t = 0  
mul = 1 # 乘数  
if num / 10 == 0:  
 t = num % 10  
else:  
 while num:  
 if i % 2 != 0:  
 t += num % 10 \* mul  
 mul = mul \* 10  
 num = num // 10  
 i += 1  
print(f't的值是{t}')**

1.7

在一个停车场内，汽车、摩托车共停了 48 辆，其中每辆汽车有 4 个轮子，每辆摩托车

有 3 个轮子，这些车共有 172 个轮子，编程输出停车场内有汽车和摩托车的数量。

**i = 0 # 汽车数量  
for i in range(1, 49):  
 if i \* 4 + (48 - i) \* 3==172:  
 print(f"汽车数量是:{i},摩托车数量是：{48-i}")**

1.8

小明今天参加了“校园歌手大赛”，评委的打分规则是去掉一个最低分和一个最高分后算出剩下分数的平均分，你能帮助小明快速的算出平均分吗？ （评委数量必须大于 2）

输入说明：首先输入一个整数 n，代表评委人数，然后输入 n 个数。请按照题目的计算规则计算出平均分然后输出。

例如输入： 6

100 90 90 80 85 95

按照题目注意计算平均分并输出： 90.0

**n = int(input('输入评委人数:'))  
point = input('输入分数:').split(' ')  
point = list(map(int, point)) # 将列表值转化为整形  
point.remove(max(point))  
point.remove(min(point))  
print(f'平均分为：{sum(point) / len(point)}')**

1.9九九乘法表、

选择乘法口诀助记功能，输出阶梯形式的 9\*9 乘法口诀表

**for i in range(1, 10):  
 for j in range(1, i+1):  
 print(f'{i}X{j}={i \* j}', end=' ')  
 print()**

1.10

输入一个字符串统计每个字符在字符串中出现的次数。

**char = input('请输入一个字符串:')  
data = []  
count = []  
for i in char:  
 if i in data:  
 count[data.index(i)] += 1  
 else:  
 data.append(i)  
 count.append(1)  
for i in range(len(data)):  
 print(f'{data[i]}:{count[i]}次')**

1.11

有 n 盏灯，编号 1～n（0<n<100）。第 1 个人把所有灯打开，第 2 个人按下所有编号为

2 的倍数的开关（这些灯将被关掉），第 3 个人按下所有编号为 3 的倍数的开关（其中关掉的灯将被打开，开着的灯将被关闭），依次类推。输入灯数和人数，输出开着的灯的编号。

比如输入：10 2 输出最后亮灯的编号：1,3,5,7,9 注意：使用循环语句实现。

**n, x = input('请依次输入灯数和人数:').split(' ')  
n = int(n)  
x = int(x)  
led = []  
# 使输入的所有灯打开  
for i in range(n + 1):  
 led.append(1)  
for i in range(2, x + 1):  
 for j in range(i, n + 1, i):  
 led[j] = -led[j]  
for i in range(1, n + 1):  
 if led[i] == 1:  
 print(i, end=',')**

1.12

冒泡排序、

原始数组：[1,9,3,7,4,2,5,0,6,8]

排序后：[0,1,2,3,4,5,6,7,8,9]；

要求：综合使用分支、循环结构语句实现，直接输出结果不计分。打印每一次运行的结果

**initial = [1, 9, 3, 7, 4, 2, 5, 0, 6, 8] # 初始值  
temp = 0 # 中间值转化  
for i in range(len(initial)):  
 for j in range(len(initial) - i-1):  
 if initial[j] > initial[j + 1]:  
 temp = initial[j]  
 initial[j] = initial[j + 1]  
 initial[j + 1] = temp  
 print(f'第{i+1}轮变化结果：{initial}')**

1.13

动物园饲养的食肉动物分大型动物和小型动物两类，规定老虎、狮子一类的大动物每次喂肉每头三斤，狐狸、山猫一类小动物每三头喂一斤。该动物园共有这两类动物 100 头，每

次需喂肉 100 斤，编程输出大、小动物的数量。

**for big in range(1, 101):  
 if 3 \* big + 1 / 3 \* (100 - big) == 100:  
 print(f'大型动物{big}，小型动物{100 - big}')**

1.14

从键盘接收一个整数 n(n>=4)，请打印一个由“\*”号组成的长度和宽度均为 n 的空心矩形。例如输入：4 你要在屏幕打印如下图形：

\*\*\*\*

\* \*

\* \*

\*\*\*\*

**n = int(input('请输入一个整数n：'))  
for i in range(n):  
 for j in range(n):  
 if i == 0 or j == 0 or i == n - 1 or j == n - 1:  
 print('\*', end='')  
 else:  
 print('', end=' ')  
 print()**

1.15

注意输出指定空心正方形。输入第一个数字为边长，第二个字符为组成图形边的字符。例如：输入

4 a （4为空心正方形的大小，a为显示的字符）

输出

aaaa

a a

a a

aaaa

**n, char = input('请依次输入边长和字符：').split(' ')  
n = int(n)  
for i in range(n):  
 for j in range(n):  
 if i == 0 or j == 0 or i == n - 1 or j == n - 1:  
 print(f'{char}', end='')  
 else:  
 print('', end=' ')  
 print()**

1.16

已知鸡和兔的总数量为 n,总腿数为 m。输入 n 和 m,依次输出鸡和兔的数目，如果无解， 则输出“No answer”(不要引号)

**n, m = input('依次输入n,m:').split(' ')  
n = int(n)  
m = int(m)  
rabbit = 0  
f = True # 判断答案是否存在  
for rabbit in range(n + 1):  
 if rabbit \* 4 + (n - rabbit) \* 2 == m:  
 f = False  
 print(f'兔的数量：{rabbit}，鸡的数量：{n - rabbit}')  
if f:  
 print('No answer')**

1.17

中国古代的《算经》记载了这样一个问题：公鸡 5 文钱 1 只，母鸡 3 文钱 1 只，小鸡 1

文钱 3 只，如果用 100 文钱买 100 只鸡，那么公鸡、母鸡和小鸡各应该买多少只呢？现在请

你编程求出所有的解，每个解输出 3 个整数，打印在一行，用空格隔开，分别代表买的公鸡、母鸡、小鸡的数量。

注意：100 文钱要正好用完。请输出所有的解，每个解占一行。

**cock, hen, chick = 0, 0, 0 # 公鸡，母鸡，小鸡  
for cock in range(100):  
 for hen in range(100):  
 for chick in range(100):  
 if cock + hen + chick == 100 and cock \* 3 + hen \* 1 + 1 / 3 \* chick == 100:  
 print(cock, hen, chick, sep=' ')**

1.18

啤酒每罐 2.3 元，饮料每罐 1.9 元。小明买了若干啤酒和饮料，一共花了 82.3 元。我们还知道他买的啤酒比饮料的数量少，请你编程计算他买了几罐啤酒。

**max\_beer = int(82.3 / 2.3)  
max\_drink = int(82.3 / 1.9)  
for beer in range(max\_beer):  
 for drink in range(max\_drink):  
 if beer \* 2.3 + drink \* 1.9 == 82.3 and beer < drink:  
 print(beer)**

1.9 折纸比高、

假设一张足够大的纸，纸张的厚度为 0.5 毫米。请问对折多少次以后，可以达到珠穆朗玛峰的高度(最新数据：8844.43 米)。请编写程序输出对折 次数。

注意：使用循环结构语句实现，直接输出结果不计分。

**sum\_height = 0.5  
t = 0 # 折纸次数  
while sum\_height < 8844.43:  
 sum\_height \*= 2  
 t += 1  
print(t-1)**

1.20

职员小 A 今天犯了一个致命的错误，他一不小心丢失了 X 项目的市场调查结果只记得一个公式 xyz+yzz=532，其中 x、y、z 均为一位数，现在请你帮忙编写一个程序求出 x、y、z 分别代表什么数。

**for x in range(10):  
 for y in range(10):  
 for z in range(10):  
 if (x \* 100 + y \* 10 + z) + (y \* 100 + z \* 10 + z) == 532:  
 print(x, y, z)**

1.21

小明带两个妹妹参加元宵灯会。别人问她们多大了，她们调皮地说：“我们俩的年龄之积是年龄之和的 6 倍”。

小明又补充说：“她们可不是双胞胎，年龄差肯定也不超过 8 岁啊。” 请你编程求出小明的较小的妹妹的年龄。

**age\_min = 0  
age\_max = 0  
for age\_min in range(1, 50):  
 for age\_max in range(1, 50):  
 if age\_min \* age\_max == (age\_max + age\_min) \* 6 and 0 < age\_max - age\_min <= 8:  
 print(age\_min)**

1.22

本月酒水的销售为 2!+4!+5!的值。n!表示 n 的阶乘，例如 3!=3×2×1=6，5!=5×4×3

×2×1=120。求这个值

**sum\_num = 0  
t = 1 # 计算每次循环的值  
for i in range(1, 6):  
 t \*= i  
 if i == 2:  
 sum\_num += t  
 if i == 4:  
 sum\_num += t  
 if i == 5:  
 sum\_num += t  
print(sum\_num)**

1.23

分别输入两个字符串 s1 和 s2 ，请问 s1 中包含多少个 s2，如果没有则输出 0。要求：使用循环。

**s1, s2 = input('按顺序输出字符串s1,s2:').split(' ')  
t = 0 # 计数器  
for i in range(len(s1) + 1):  
 if s2 == s1[i:i + len(s2)]:  
 t += 1  
print(t)**

1.24

小米打算把图标设计成下面这样，但是他不知道几层最合适，于是想写个程序，打印出不同层的图标，请你帮帮他。

—

———

—————

———————

注意：输入 n，打印 n 层高度的图标，如图是 4 层。

**n = int(input('请输入n:'))  
for i in range(n):  
 for j in range(n - i):  
 print(' ', end='')  
 for j in range(1 + 2 \* i):  
 print('-', end='')  
 print()**

1.26

Lee 的老家住在工业区，日耗电量非常大。

今年 7 月，传来了不幸的消息，政府要在 7、8 月对该区进行拉闸限电。政府决定从 7

月 1 日起停电，然后隔一天到 7 月 3 日再停电，再隔两天到 7 月 6 日停电，依次下去，每次都比上一次长一天。

Lee 想知道自己到家后到底要经历多少天倒霉的停电。请编写程序帮他算一算。

注意：从键盘输入放假日期、开学日期，日期限定在 7、8 月份，且开学日期大于放假日期，然后在屏幕上输出停电天数。

提示：可以用数组标记停电的日期。

**date7 = [f'7-{i}' for i in range(1, 32)]**

**date8 = [f'8-{i}' for i in range(1, 32)]**

**date\_time = date7 + date8**

**date\_dict = {}**

**start = 0**

**s = 1**

**for index, i in enumerate(date\_time):**

**date\_dict[i] = 0**

**if start == index:**

**date\_dict[i] = 1**

**s += 1**

**start = start + s**

**print(date\_dict)**

**date\_list = list(date\_dict.keys())**

**start\_date = input('请输入开始的时间')**

**stop\_date = input('请输入截至的时间')**

**start\_index = date\_list.index(start\_date)**

**stop\_index = date\_list.index(stop\_date)**

**power\_cut = []**

**for i in range(start\_index, stop\_index + 1):**

**if date\_dict[date\_list[i]] == 1:**

**power\_cut.append(date\_list[i])**

**print(power\_cut)**

**print(len(power\_cut))**

1.27

物园里新来了两只骆驼，那么你能计算出它们年龄的最小公倍数么？ 从键盘输入两个整数，输出两个整数的最小公倍数。

**age\_a, age\_b = input('依次输入两个整数：').split(' ')  
age\_a = int(age\_a)  
age\_b = int(age\_b)  
temp = 0 # 替换值  
if age\_b > age\_a:  
 temp = age\_b  
 age\_b = age\_a  
 age\_a = temp  
i = age\_a # 计数器  
while 1:  
 if i % age\_a == 0 and i % age\_b == 0:  
 print(i)  
 break  
 i \*= 2**

1.28

问题:使用给定的整数n，编写一个程序生成一个包含(i, i\*i)的字典，该字典包含1到n之间的整数(两者都包含)。然后程序应该打印字典。

假设向程序提供以下输入:8

则输出为:

{1:1，2:4，3:9，4:16，5:25，6:36，,7:49，8:64}

提示:在为问题提供输入数据的情况下，应该假设它是控制台输入。考虑使用dict类型()

**n = int(input('请输入整数n:'))  
dict\_1 = {}  
for i in range(1, n + 1):  
 dict\_1[i] = i \*\* 2  
print(dict\_1)**

1.29

编写一个接受句子的程序，并计算大写字母和小写字母的数量。  
假设为程序提供了以下输入：  
Hello world!  
然后，输出应该是：  
大写实例 1  
小写实例 9

**char = input('请输入字符串:')  
small = 0  
big = 0  
for i in char:  
 if i.isupper():  
 big += 1  
 elif i.islower():  
 small += 1  
print(f'大写实例 {big}')  
print(f'小写实例 {small}')**

面向对象

1.1

问题:定义一个至少有两个方法的类:

getString:从控制台输入获取字符串

 printString::打印大写母的字符串。  
还请包含简单的测试函数来测试类方法。  
提示:使用\_init\_\_方法构造一些参数

**class ChangeBig:  
 def \_\_init\_\_(self):  
 char = self.getString()  
 self.printString(char)  
  
 def printString(self, char):  
 for i in char:  
 if i.isupper():  
 print(i, end='')  
  
 def getString(self):  
 char = input('请输入字符串:')  
 return char  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 w = ChangeBig()**

练习进阶

1.1

我们都知道：1+2+3+ ... + 49 = 1225。现在要求你把其中两个不相邻的加号变成乘号， 使得结果为 2015。

例如： 1+2+3+...+10\*11+12+...+27\*28+29+...+49 = 2015 就是符合要求的答案。

请你寻找所有可能的答案，并把前面的两个数字输出，如上面的就是输出（10 27）

**for i in range(1, 50):  
 for j in range(i + 1, 50):  
 for k in range(1, 50):  
 if i == k: # 第一个乘数  
 sum\_num = sum\_num + k \* (k + 1)  
 first\_num = i  
 elif j == k: # 第二个乘数  
 sum\_num = sum\_num + k \* (k + 1)  
 second\_num = j  
 else:  
 if k == i + 1 or k == j + 1:  
 continue  
 else:  
 sum\_num += k  
 if sum\_num == 2015 and second\_num - first\_num != 1:  
 print(first\_num, second\_num)  
 sum\_num = 0**

1.2

输入一个整数 n，计算并输出他的阶乘。

注意：定义一个函数(或方法)，用于求阶乘的值。

在主函数(或主方法)中调用该递归函数(或方法)，求出 5 的阶乘，并输出结果。

**def get\_factorial(n):  
 if n == 1:  
 return 1  
 else:  
 return get\_factorial(n - 1) \* n  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 n = int(input('请输入整数n:'))  
 print(get\_factorial(n))  
 print(f'5的阶乘{get\_factorial(5)}')**

1.3

由于中国结的形状是菱形图案，所以现在公司需要设计一个打印菱形的方法。从键盘输入一个整数 N，打印出有 N\*2-1 行的菱形。

例如输入整数 4，则屏幕输出如下菱形。

\*

\*\*\*

\*\*\*\*\*

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

\*\*\*\*\*

\*\*\*

\*

现要求输入整数为 7，在屏幕中输出相应的菱形。要求：用循环结构语句实现。

n = int(input('请输入整数n:'))  
# 输出上半部分  
**for i in range(n):  
 for j in range(n - i):  
 print(' ', end='')  
 for j in range(1 + 2 \* i):  
 print('\*', end='')  
 print()  
# 输出下半部分  
for i in range(1, n):  
 for j in range(i + 1):  
 print(' ', end='')  
 for j in range(1 + 2 \* (n - i - 1)):  
 print('\*', end='')  
 print()**

1.4

编写程序输出 2~99 之间的同构数。同构数是指这个数为该数平方的尾数，例如 5 的平方为 25，6 的平方为 36，25 的平方为 625，则 5、6、25 都为同构数。

注意：调用带有一个输入参数的函数(或方法)实现，此函数(或方法)用于判断某个整数是否为同构数，输入参数为一个整型参数，返回值为布尔型（是否为同构数）。

**def judge\_same(n):  
 if n < 10:  
 if (n \* n) % 10 == n:  
 return True  
 else:  
 return False  
 else:  
 if (n \* n) % 100 == n:  
 return True  
 else:  
 return False  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 n = int(input('请输入整数n:'))  
 if judge\_same(n):  
 print('YES')  
 else:  
 print('NO')**

1.5

分析下列数据的规律，编写程序完成如下所示的输出。

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

要求：使用循环结构语句实现。

**list1 = [1]**

**for i in range(6): # 打印六层**

**print(list1)**

**list1.append(1)**

**list2 = list1.copy()**

**for j, s in enumerate(range(len(list1) - 2), start=1):**

**list1[j] = list2[j] + list2[j - 1]**

1.6

在列表中 20 个不同的整数，找出其中最小的数，将它与第 1 个输入的数交换位置之后输出这些数

**data = input('请依次输入20个数：').split(' ')  
data = list(map(int, data)) # 将列表值转化为整形  
temp = 0 # 替换值  
min\_num = data[0]  
for i in data:  
 if i < min\_num:  
 min\_num = i  
temp = data[data.index(min\_num)]  
data[data.index(min\_num)] = data[0]  
data[0] = temp  
print(data)**

1.7

美国数学家维纳(N.Wiener)智力早熟，11 岁就上了大学。他曾在 1935~1936 年应邀来中国清华大学讲学。

一次，他参加某个重要会议，年轻的脸孔引人注目。于是有人询问他的年龄，他回答说：“我年龄的立方是个 4 位数。我年龄的 4 次方是个 6 位数。这 10 个数字正好包含了从 0 到9 这 10 个数字，每个都恰好出现 1 次。”请你编程计算，他当时到底有多年轻。

注意：使用循环实现，输出他的年龄在一行。

**def split\_int(num): # 差分数字  
 num = str(num)  
 data = []  
 data.extend(num)  
 data = list(map(int, data))  
 return data  
  
  
def check\_same(data): # 检测是否有重复（有True，没有False）  
 t = 0 # 计数器  
 for i in data:  
 for j in data:  
 if i == j:  
 t += 1  
 if t > 1:  
 return True  
 t = 0  
 else:  
 return False  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 data = []  
 for age in range(100):  
 if age \*\* 3 // 1000 != 0: # 我年龄的立方是个 4 位数  
 if age \*\* 4 // 100000 != 0: # 我年龄的 4 次方是个 6 位数  
 data = split\_int(age \*\* 3) + split\_int(age \*\* 4)  
 if sum(data) == 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 and 1 - check\_same(data):  
 print(age)  
  
  
# 第二种方法（数字取余法）  
def check\_same(data): # 检测是否有重复（有True，没有False）  
 t = 0 # 计数器  
 for i in data:  
 for j in data:  
 if i == j:  
 t += 1  
 if t > 1:  
 return True  
 t = 0  
 else:  
 return False  
  
  
data = []  
for age in range(100):  
 if age \*\* 3 // 1000 != 0: # 我年龄的立方是个 4 位数  
 if age \*\* 4 // 100000 != 0: # 我年龄的 4 次方是个 6 位数  
 # 这 10 个数字正好包含了从 0 到9 这 10 个数字  
 data = [age \*\* 3 % 10, age \*\* 3 % 100 // 10, age \*\* 3 % 1000 // 100, age \*\* 3 // 1000,  
 age \*\* 4 % 10, age \*\* 4 % 100 // 10, age \*\* 4 % 1000 // 100,  
 age \*\* 4 % 10000 // 1000, age \*\* 4 // 10000 % 10, age \*\* 4 // 100000]  
 if sum(data) == 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 and 1 - check\_same(data):** **print(age)**

1.8

输入整数 a，输出结果 s，其中 s 与 a 的关系是：s=a+aa+aaa+aaaa+aa...a，最后为 a 个 a。例如 a=2 时，s=2+22=24。

注意：①使用循环结构语句实现。②a 由键盘输入，且 2 ≤ a ≤9。

**a = int(input('输入整数a:'))  
sum\_a = 0  
s = 0  
for i in range(a):  
 for j in range(i+1):  
 sum\_a += 10 \*\* j \* a  
 s += sum\_a  
 sum\_a = 0  
print(s)**

1.9

判断一个整数是否是回文数。

(1)题目分析：回文数是指正序(从左向右)和倒序(从右向左)读都是一样的整数。

**def split\_int(num): # 差分数字  
 num = str(num)  
 data = []  
 for i in num:  
 data.append(i)  
 data = list(map(int, data))  
 return data  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 data = split\_int(input('请输入一个整数：'))  
 for i in range(len(data) // 2):  
 if data[i] != data[len(data) - 1 - i]:  
 print('NO')  
 break  
 else:  
 print('YES')**

难度增加

1.1

一个猪圈里有2只刚满2岁的猪，

每只猪只能活4年，每只猪第四年还可以生一次崽

2岁后每2只猪每年可以生4头小猪（假设小猪性别比例总是1：1），

问n年后猪圈里有几只猪？

**pig\_num = [2]  
n = int(input('请输入n年：'))  
for i in range(n): # 每一年进行增改  
 for j in range(len(pig\_num)):  
 pig\_num[j] += 1 # 每一年增加一岁  
 if pig\_num[j] > 2: # 2岁后每2只猪每年可以生4头小猪(因为小猪性别比例总是1：1，所以两只小猪比作1只)  
 pig\_num += [1, 1]  
 if pig\_num[j] == 4: # 四年后猪死了  
 del pig\_num[j]  
print(len(pig\_num) \* 2)**

1.2

题：使用生成器定义一个类，该生成器可以在给定范围0和n之间迭代可被7整除的数字。

提示：考虑使用yield。

**def foo(n):  
 i = 0  
 while i < n:  
 j = i  
 i = i + 1  
 if j % 7 == 0:  
 yield j  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 n = int(input('请输入正整数n:'))  
 for i in foo(n):  
 print(i)**

1.3

给定 n 个非负整数表示每个宽度为 1 的柱子的高度图，计算按此排列的柱子，下雨之后能接多少雨水。

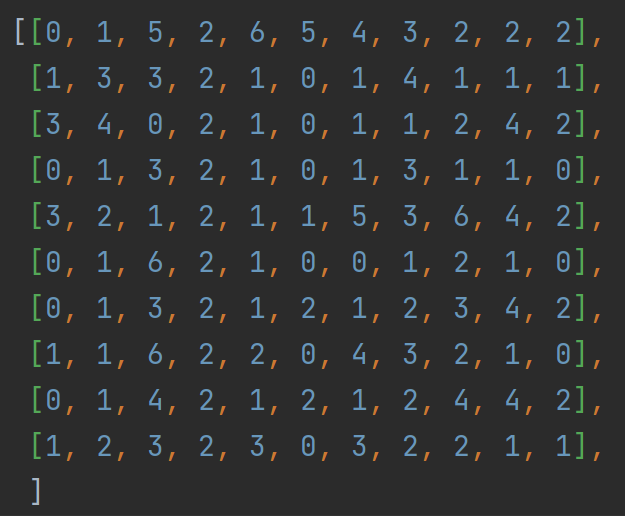


上面是由[数组](https://so.csdn.net/so/search?q=%E6%95%B0%E7%BB%84&spm=1001.2101.3001.7020) [0,1,0,2,1,0,1,3,2,1,2,1] 表示的高度图，在这种情况下，可以接 6 个单位的雨水（蓝色部分表示雨水）。

**n = input('请依次输入柱子高度非负整数n:').split(' ')  
n = list(map(int, n)) # 转化为整数  
  
  
# n = [0, 1, 0, 2, 1, 0, 1, 3, 2, 1, 2, 1]  
# 判断该雨滴是否能被接收  
  
def judge\_is\_rain(transverse, data): # 该点的横坐标，纵坐标，值  
 left = False  
 right = False  
 #  
 for i in range(len(n)):  
 if n[i] > data and i < transverse: # 判断左边是否有大于该点的值  
 left = True  
 if n[i] > data and i > transverse: # 判断右边是否有大于该点的值  
 right = True  
 if left and right and data == 0:  
 return True  
 else:  
 return False  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 t = 0 # 计数器  
 sum\_rain = 0 # 雨滴总数  
 while t != len(n):  
 t = 0  
 for i in range(len(n)):  
 if judge\_is\_rain(i, n[i]): # 获取雨滴  
 sum\_rain += 1 # 总雨滴数+1  
 n[i] += 1 # 补充该与滴的位置  
 else:  
 t += 1 # 不满足的层数+1  
 else: # 每判断一层后层数减1  
 for i in range(len(n)):  
 n[i] = n[i] - 1  
print(sum\_rain)**

1.3 改进版

接雨水从2D变成3D 长宽12 简单示例：



在这种情况下，可以接 多少个单位的雨水呢？

**pillar = [[0, 1, 5, 2, 6, 5, 4, 3, 2, 2, 2],  
 [1, 3, 3, 2, 1, 0, 1, 4, 1, 1, 1],  
 [3, 4, 0, 2, 1, 0, 1, 1, 2, 4, 2],  
 [0, 1, 3, 2, 1, 0, 1, 3, 1, 1, 0],  
 [3, 2, 1, 2, 1, 1, 5, 3, 6, 4, 2],  
 [0, 1, 6, 2, 1, 0, 0, 1, 2, 1, 0],  
 [0, 1, 3, 2, 1, 2, 1, 2, 3, 4, 2],  
 [1, 1, 6, 2, 2, 0, 4, 3, 2, 1, 0],  
 [0, 1, 4, 2, 1, 2, 1, 2, 4, 4, 2],  
 [1, 2, 3, 2, 3, 0, 3, 2, 2, 1, 1], ] # 所有的柱子数量  
m = 10 # 长  
n = 11 # 宽  
# 判断这个雨滴是否在可以收集  
def judge\_is\_rain(transverse, longitudinal, data): # 该点的横坐标，纵坐标，值  
 left, right, before, behind = False, False, False, False  
 # 判断左边是否有峰值  
 for i in range(longitudinal - 1, -1, -1):  
 if pillar[transverse][i] > 0:  
 left = True  
 # 判断右边是否有峰值  
 for i in range(longitudinal + 1, n):  
 if pillar[transverse][i] > 0:  
 right = True  
 # 判断上方是否有峰值  
 for j in range(transverse - 1, -1, -1):  
 if pillar[j][longitudinal] > 0:  
 before = True  
 # 判断下方是否有峰值  
 for j in range(transverse + 1, m):  
 if pillar[j][longitudinal] > 0:  
 behind = True  
 if left and right and before and behind and data == 0:  
 return True  
 else:  
 return False  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 sum\_rain = 0 # 总水滴量  
 t = 0 # 计数器  
 ii = 0  
 # print(len(pillar))  
 # print(pillar[0][3])  
 while t != n \* m:  
 ii += 1  
 t = 0  
 for i in range(m):  
 for j in range(n):  
 # print(f'piller[{i}][{j}]={pillar[i][j]}')  
 if judge\_is\_rain(i, j, pillar[i][j]):  
 # print(i, j, pillar[i][j], ii)  
 pillar[i][j] += 1  
 sum\_rain += 1 # 总水滴量  
 elif pillar[i][j] < 0:  
 t += 1  
 else:  
 # 所有层数减小1  
 for i in range(m):  
 for j in range(n):  
 pillar[i][j] -= 1  
 print(sum\_rain)**