1. 小明单位发了100元的购物卡，小明到超市买三类洗化用品：洗发水（15元）、香皂（2元）、牙刷（5元）。要把100元正好花掉，可有哪些购买组合？

money=100

n=money//15

for i in range(n,-1,-1):

m=(money-i\*15)//5

for j in range(m,-1,-1):

k=(money-i\*15-j\*5)//2

if (money-i\*15-j\*5)%2==0:

print('可选择的购买组合:\t\t购买洗发水 {} 瓶，香皂 {} 块，牙刷 {}个。'.format(i,j,k))

2设计一个猜数游戏。首先由计算机产生一个[1,100]之间的随机整数，然后由用户猜测所产生的随机数。根据用户猜测的情况给出不同提示，如猜测的数大于产生的数，则显示“High”，小于则显示“Low”，等于则显示“You won !”，游戏结束。用户最多可以猜7次，如果7次均未猜中，则显示“You lost !”，并给出正确答案，游戏结束。游戏结束后，询问用户是否继续游戏，选择“Y”则开始一轮新的猜数游戏；选择“N”则退出游戏。

chose = 'y'

while chose=='Y' or chose=='y':

import random

num = random.randint(1,100)

def judge(b):

if b == num:

return 1

else:

return 0

for i in range(1,8):

b=eval(input('请输入您第{}次所猜的整数：'.format(i)))

if judge(b)==1:

print("You won !")

break

elif b > num:

print("high")

elif b < num:

print("low")

if judge(b)==0:

print("You lost !")

chose=input('请输入Y(y)继续进行游戏，N(n)退出游戏：')

while chose != 'Y' and chose != 'y' and chose != 'N' and chose != 'n':

print('输入有误，请重新输入Y(y)继续进行游戏，N(n)退出游戏：',end = '')

chose=input()

3建立1个包含10个字符的字符串，并根据键盘输入的数字n输出字符串中的第n个字符。当n值超过字符串的索引时，自动转为输出字符串中的最后1个字符。

4，编写函数，该函数可以输入任意多个数，函数返回输出所有输入参数的最大值、最小值和平均值。

5 一个人赶着鸭子去每个村庄卖，每经过一个村子卖去所赶鸭子的一半又一只。这样他经过了七个村子后还剩两只鸭子，问他出发时共赶多少只鸭子？

6 输入一个日期，判断这天是星期几？

7，数字加密游戏：编程程序，从键盘任意输入1个4位数，将该数字中的每位数与7相乘，然后取乘积结果的个位数对该数字进行替换，最后得到1个新的4位数。

8简述：这里有四个数字，分别是：1、2、3、4

提问：能组成多少个互不相同且无重复数字的三位数？各是多少？

for i in range(1, 5):

for j in range(1, 5):

for k in range(1, 5):

if (i != k) and (i != j) and (j != k):

print(i, j, k)

9，Python完全平方数，编程练习题实例三

简述：一个整数，它加上100和加上268后都是一个完全平方数

提问：请问该数是多少？ Python解题思路分析：在10000以内判断，将该数加上100后再开方，加上268后再开方，如果开方后的结果满足如下条件，即是结果。 Python完全平方数，

import math

for i in range(10000):

#转化为整型值

x = int(math.sqrt(i + 100))

y = int(math.sqrt(i + 268))

if(x \* x == i + 100) and (y \* y == i + 268):

print (i)

10，Python整数顺序排列，编程练习题实例五

整数顺序排列问题简述：任意三个整数类型，x、y、z

提问：要求把这三个数，按照由小到大的顺序输出 Python解题思路分析：

首先，要想方法把最小的数放到x位上，之后将x与y进行比较； 如果x>y的话，就将x与y的值进行交换； 然后再用x与z进行比较，如果x>z则将x与z的值进行交换，这样能使x最小。 Python整数顺序排列

11，实现斐波那契数列，斐波那契数列（Fibonacci sequence），又称黄金分割数列、因数学家列昂纳多·斐波那契以兔子繁殖为例子而引入，故又称为“兔子数列”，指的是这样一个数列：1、1、2、3、5、8、13、21、34、在数学上，斐波纳契数列以如下被以递归的方法定义。输入一个长度，输出指定个数的斐波那契数列

12，数据类型判断，要求创建一个函数，传入一个随机的数据，判断他的类型：

如传入一个列表类型，输出：这个数据是一个列表类型

13一辆卡车违反交通规则,撞人后逃跑.现场有三人目击事件,但都没有记住车号,只记下车号的一些特征.

甲说：牌照的前两位数字是相同的；

乙说：牌照的后两位数字是相同的,但与前两位不同；

丙是数学家,他说：四位的车号所构成的数字正好等于某一个整数的平方.

请根据以上线索求出车号.

提示：四位整数中的完全平方数 的取值范围：32的平方—99的平方.

14有一个二维的正整数数组（自己随机创建一个），对每一个数据的周边3-8个数字进行差值计算，当发现周围和他的差值（相减）大于10，那么将它和那个差值数都变成-1

15 题目名称：批阅奏章  
某朝皇帝有大臣n名（1<=n<=1000），分别编号大臣1~n。某日皇帝身体抱恙，奏章堆积如山无法及时一一批阅，便命身旁內侍帮他把奏章按指定顺序排序后再阅。于是皇帝亲自挑选了几个值得信赖的重臣并排好序，要求把他们的奏章按排好的顺序放到前面，其他的按照编号升序排列即可。现在要求你写一个程序来帮皇上解决这个问题，即已知奏章总数和顺序、钦点重臣的排列顺序，求得皇帝查阅奏章的顺序。  
输入描述：  
第一行输入两个整数p（1<=p<=5000）和q，其中p表示堆积奏章的总数、q表示皇帝钦点重臣数  
第二行输入p个数，表示所有按呈递顺序递上来的奏章来自于哪个大臣（大臣编号）  
第三行输入q个数，表示皇帝钦点并排好序的重臣编号  
输出描述：  
输出奏章按指定顺序排好序后，皇帝按大臣编号批阅的顺序  
输入样例：  
5 3  
5 4 3 2 1  
3 5 4  
输出样例：  
3 5 4 1 2  
答案：

line1 = list(map(int,input('ddd').split()))

p = line1[0]

q = line1[1]

article\_author = list(map(int,input('ddd').split()))

persons = list(map(int,input('ddd').split()))

result =[]

article\_author.sort()

for p1 in persons:

for a1 in range(0,len(article\_author)):

if article\_author[a1] == p1:

result.append(p1)

article\_author[a1] = 0

for a1 in article\_author:

if a1!=0:

result.append(a1)

print\_text = list(map(str,result))

print(" ".join(print\_text))

16 题目名称：报价  
题目描述  
给定某股票每日的报价和一个目标值，请在所有报价中找出和为目标值的那两天的报价，并打印出对应的报价。  
假设每种输入只会对应一个答案，且每日的报价不会重复。  
你需要按报价从小到大的顺序打印答案。

输入描述：  
输入：第一行是某股票每日的报价，这些报价是正整数且用空格相隔，例如：17 20 33  
第二行是目标值，例如：37  
输出描述：  
输出：对应的报价，报价之间用空格相隔，例如：17 20  
示例  
示例1  
输入  
17 20 33  
37  
复制  
输出  
17 20  
复制  
答案：

list1=list(map(int,input().split()))

goal=int(input())

list1.sort()

for i in list1:

if (goal-i) in list1:

print(str(i)+" "+str(goal-i))

break

17 题目名称：字符串查找和比较(pass)  
题目描述  
写函数实现如下功能，给定字符串A和B,输出A和B中的最长公共子串。比如A=“aocdfe” B=“pmcdfa” 则输出"cdf"。  
输入描述：  
输入待处理的两个字符串 str1，str2  
输出描述：  
找出两个字符串最长的公共子串  
示例 示例1  
输入  
aocdfe  
pmcdfa  
输出  
Cdf  
答案：

A = str(input(""))

B = str(input(""))

if len(A)>len(B):

A,B = B,A

out\_put=[]

for i in range(len(A),0,-1):

for j in range(0,len(A)-i+1):

if A[j:j+i] in B:

print(A[j:j+i])

out\_put.append(A[j:j+i])

if out\_put:

break

18 题目名称：尼姆博弈  
题目描述  
你和你的朋友，两个人一起玩 Nim 游戏：  
桌子上有一堆石头。  
你们轮流进行自己的回合，你作为先手。  
每一回合，轮到的人拿掉 1 - 3 块石头。  
拿掉最后一块石头的人就是获胜者。  
假设你们每一步都是最优解。请编写一个函数，来判断你是否可以在给定石头数量为 n 的情况下赢得游戏。如果可以赢，返回 true；否则，返回 false 。  
输入描述：  
整数n  
输出描述：  
true或false  
示例  
示例1  
输入  
4  
复制  
输出  
False  
答案：

num = int(input())

if num % 4 == 0:

print("false")

else:

print("true")

19 题目名称：罗马数字转整数(pass)  
题目描述  
罗马数字包含以下七种字符: I， V， X， L，C，D 和 M。

例如， 罗马数字 2 写做 II ，即为两个并列的 1 。12 写做 XII ，即为 X + II 。 27 写做 XXVII, 即为 XX + V + II 。  
通常情况下，罗马数字中小的数字在大的数字的右边。但也存在特例，例如 4 不写做 IIII，而是 IV。数字 1 在数字 5 的左边，所表示的数等于大数 5 减小数 1 得到的数值 4 。同样地，数字 9 表示为 IX。这个特殊的规则只适用于以下六种情况：

I 可以放在 V (5) 和 X (10) 的左边，来表示 4 和 9。  
X 可以放在 L (50) 和 C (100) 的左边，来表示 40 和 90。  
C 可以放在 D (500) 和 M (1000) 的左边，来表示 400 和 900。  
给定一个罗马数字，将其转换成整数。  
输入描述：  
罗马数字  
输出描述：  
转换后的整数  
示例  
示例1  
输入 III  
输出 3  
答案：

src = input()

n\_value = {"Z":0,"I":1,"V":5,"X":10,"L":50,"C":100,"D":500,"M":1000}

pre\_value = {"I":"Z","V":"I","X":"I","L":"X","C":"X","D":"C","M":"C"}

result = 0

last\_char = ""

for \_c in src:

result+= n\_value[\_c]

if last\_char == pre\_value[\_c]:

result-= n\_value[last\_char] \* 2

last\_char = \_c

print(result)

20 题目名称：汉诺塔  
题目描述  
有三个立柱A、B、C。A柱上穿有大小不等的圆盘N个，较大的圆盘在下，较小的圆盘在上。要求把A柱上的圆盘全部移到C柱上，保持大盘在下、小盘在上的规律（可借助B柱）。每次移动只能把一个柱子最上面的圆盘移到另一个柱子的最上面。请输出移动过程。

输入描述：  
输入一个整数n

输出描述：  
输出移动过程

示例  
示例1  
输入  
3  
复制  
输出  
a->c  
a->b  
c->b  
a->c  
b->a  
b->c  
a->c

def han\_nuo(n,src,tmp,dest):

if n <= 0:

return

if n == 1:

print(src + "->" + dest)

return

han\_nuo(n-1,src,dest,tmp)

print(src + "->" + dest)

han\_nuo(n-1,tmp,src,dest)

num = int(input())

han\_nuo(num,"a","b","c")

21 题目名称：丑数  
题目描述  
给你一个整数 n ，请你判断 n 是否为 丑数 。如果是，返回 true ；否则，返回 false 。  
丑数 就是只包含质因数 2、3 和/或 5 的正整数。(1通常被视为丑数)  
输入描述：  
整数n  
输出描述：true或false  
示例 示例1  
输入6  
输出true  
答案：

m=int(input())

def cnum(n):

while True:

if n <= 0:

return False

elif n==1 or n==2 or n==3 or n==5:

return True

elif n % 2 ==0:

n = n/2

elif n % 3 ==0:

n = n/3

elif n % 5 ==0:

n = n/5

else:

return False

f=cnum(m)

if f == True:

print("true")

else:

print("false")

22 克拉兹猜想：任取一正整数，如果是偶数，将其除以2。如果是奇数，将其乘以3再加1，然后重复这个过程，最后结果都会陷入4 2 1 的循环。  
比如序列：13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1  
请通过编程实现，当4,2,1重复第二次的时候，结束循环。请输出以下3个序列：  
[13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1]  
[31, 94, 47, 142, 71, 214, 107, 322, 161, 484, 242, 121, 364, 182, 91, 274, 137, 412, 206, 103, 310, 155, 466, 233, 700, 350, 175, 526, 263, 790, 395, 1186, 593, 1780, 890, 445, 1336, 668, 334, 167, 502, 251, 754, 377, 1132, 566, 283, 850, 425, 1276, 638, 319, 958, 479, 1438, 719, 2158, 1079, 3238, 1619, 4858, 2429, 7288, 3644, 1822, 911, 2734, 1367, 4102, 2051, 6154, 3077, 9232, 4616, 2308, 1154, 577, 1732, 866, 433, 1300, 650, 325, 976, 488, 244, 122, 61, 184, 92, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1]  
[101, 304, 152, 76, 38, 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1]

def collatz(number):

res=number%2

if res==0:

return number//2

else:

return number\*3+1

num=13

in\_list=[13,31,101]

for num in in\_list:

jg\_list = [num]

while True:

num=collatz(num)

jg\_list.append(num)

if jg\_list.count(1)==2:

break

print(jg\_list)