1.求整数的和与均值(P040)：读入n个整数，求它们的和与均值。

解答：

n = int(input())

sum = 0

s = input()

if " " in s: #或 len(s.split()) > 1

nums = s.split()

for i in range(n):

sum += int(nums[i])

else:

sum += int(s)

for i in range(n-1):

sum += int(input())

print(sum,'%.5f'%(sum/n))

2.整数序列的元素最大跨度值(P041):给定一个长度为n的非负整数序列，请计算序列的最大跨度值（最大跨度值 = 最大值减去最小值）。

解答：

n = int(input())

lst = input().split()

minX = maxX = int(lst[0]) #读入的第一个数当作当前最大值和最小值

for i in range(n-1):

x = int(lst[i+1])

minX = min(x,minX)

maxX = max(x,maxX)

print(maxX-minX)

3.奥运奖牌计数(P043)：A国的运动员参与了n天的奥运比赛项目，已知该国每一天获得的金、银、铜牌数目。现在要统计一下A国所获得的金、银、铜牌数目及总奖牌数。

解答：

n = int(input())

gold = silver = bronze = 0

for i in range(n):

num = input().split()

gold += int(num[0])

silver += int(num[1])

bronze += int(num[2])

print(gold,silver,bronze,gold+silver+bronze)

4.鸡尾酒疗法(P044)：现在要通过临床对照实验的方式验证各种新疗法是比鸡尾酒疗法疗效更好。假设鸡尾酒疗法的有效率为x，新疗法的有效率为y，如果y-x大于5%，则效果更好，如果x-y大于5%，则效果更差，否则称为效果差不多。给出n组临床对照实验的疗效数据，每组数据由病人数和痊愈人数两个整数构成，其中第一组采用鸡尾酒疗法，其他n-1组为各种不同的新疗法。请写程序判定各种新疗法效果相比鸡尾酒疗法如何。

解答：

n = int(input())

lst = input().split()

x = int(lst[1])/int(lst[0])

for i in range(n-1):

lst = input().split()

y = int(lst[1])/int(lst[0])

if y - x > 0.05:

print("better")

elif x - y > 0.05:

print("worse")

else:

print("same")

5.角谷猜想：所谓角谷猜想，是指对于任意一个正整数，如果是奇数，则乘3加1，如果是偶数，则除以2，得到的结果再按照上述规则重复处理，最终总能够得到1。如，假定初始整数为5，计算过程分别为16、8、4、2、1。程序要求输入一个整数，将经过处理得到1的过程输出来。

解答：

n = int(input())

while n != 1:

if n % 2:

print(str(n) + "\*3+1=" + str(n\*3+1))

n = n \* 3 + 1

else:

print(str(n) + "/2=" + str(n//2))

n //= 2

print("End")

6.正常血压(P045)：监护室每小时测量一次病人的血压，若收缩压在90 - 140之间并且舒张压在60 - 90之间（包含端点值）则称之为正常，现给出某病人若干次测量的血压值，计算病人保持正常血压的最长小时数。

解答：

n = int(input())

total = 0

maxTotal = 0

for i in range(n):

s = input().split()

high ,low = int(s[0]),int(s[1])

if 90 <= high <= 140 and 60 <= low <= 90:

total += 1

if total > maxTotal:

maxTotal = total

else:

total = 0

print(maxTotal)

7.数字反转(P046)：给定一个整数（可以是负的），请将该数各个位上数字反转得到一个新数。新数不得有多余的前导0。

解答：

解法一：数学解法

a = int(input())

b = abs(a) #取绝对值

c = 0

while b > 0:

#依靠除法和取模，依次取出b的个位数，十位数....加权累加到c上

c = c \* 10 + (b % 10)

b //= 10

#此时c就是b的反转的结果

if a < 0:

print(-c)

else:

print(c)

解法二：纯字符串手工解法

s = input()

L = len(s)

t = ""

for i in range(L-1,-1,-1): #从后往前取出s的字符拼接起来

t += s[i]

i = 0

while t[i] == "0" and i < L-1:

#跳过前导0 ，但个位的那个0要保留。

i += 1

#此时t[i]不是"0"，或是个位的那个"0"

if s[0] == "-": #输入是负数

print("-" + t[i:-1]) # :-1是因为要把尾巴的负号去掉

else:

print(t[i:])

解法三：巧用 int 和字符串切片

#a[x:y:z]的意思，是从a[x]取到a[y](a[y]不算），每隔z个取一个，最后拼起来.z为负数代表倒着取

#x,y可以省略。x,y全省略表示从头取到尾或从尾取到头

s = input()

if "-" in s: #也可以 int(s) < 0， s[0] == "-"

s = "-" + s[-1:0:-1]

else:

s = s[::-1] #倒着从尾取到头

print(int(s))

8.数字统计：请统计某个给定范围[L, R]的所有整数中，数字2出现的次数。比如给定范围[2, 22]，数字2在数2中出现了1次，在数12中出现1次，在数20中出现1次，在数21中出现1次，在数22中出现2次，所以数字2在该范围内一共出现了6次。

解答：

s = input().split()

L,R = int(s[0]),int(s[1])

total = 0

for i in range(L,R+1):

for x in str(i):

total += (x == '2')

print(total)

9.编写add函数（非OJ题目）：编写函数add，使得表达式

add(a1)(a2)......(an)()

返回值是a1+a2+......+an。提示：让add(x)返回一个闭包，该闭包接受的参数个数可变。

解答：

def add(x, \*args):

if len(args)==0:

return x

else:

for i in range(len(args)):

x = x + args[i]

return x

10.质数生成器函数（非OJ题目）：写一个生成器函数primes生成质数，

此生成器函数为 primes(begin, end)

如: [x for x in primes(10, 20)] 将得到列表

[11, 13, 17, 19]

解答：

def primes(begin, end):

result = []

if begin <= 2:

begin = 3

result.append(2)

else:

if begin%2==0:

begin += 1

if end < 2:

return None

for i in range(begin, end+1, 2):

ok = True

for k in range(3, i, 2):

if i%k == 0:

ok = False

break

if k\*k > i:

break

if ok:

result.append(i)

return result

1.求最大公约数问题(P053)：给定两个正整数，用辗转相除法求它们的最大公约数。

解答：

解法一，递归:

def gcd(a,b):

if a % b == 0:

return b

else:

return gcd(b,a%b)

lst = input().split()

a,b = int(lst[0]),int(lst[1])

print(gcd(a,b))

解法二，迭代

lst = input().split()

a,b = int(lst[0]),int(lst[1])

while a % b != 0:

a,b = b,a%b

print(b)

2.递归复习法(P054):学渣复习期末考试，要用递归复习法，即当他复习知识点k的时候，他发现理解知识点k必须先理解知识点k-1和知识点k-2，于是他先去学习知识点k-1和知识点k-2，当他复习知识点k-1的时候，又发现理解知识点k-1必须先理解知识点k-2与知识点k-3，又得先去复习知识点k-2和知识点k-3。已知复习每个知识点所需的时间,问要多少时间才能复习完知识点n。

解答：

def mytime(k):

if k < 3:

return k

return k + mytime(k-1) + mytime(k-2)

n = int(input())

for i in range(n):

k = int(input())

print(mytime(k))

3.全排列(P057)：给定一个由不同的小写字母组成的字符串，输出这个字符串的所有全排列。 我们假设对于小写字母有'a' < 'b' < ... < 'y' < 'z'，而且给定的字符串中的字母已经按照从小到大的顺序排列。（提示：和N皇后问题一样，要使用列表）

解答：

s = input()

lst = list(s)

lst.sort()

n = len(lst)

result = [0] \* n

def permutation(i):

if i == n: # 所有位置都填上了不重复的字符

for x in result:

print(x,end="")

print("")

return

for x in range(n): # 从小到大逐个考察当前位置可以填充的字符

ok = True

for y in range(i): # 之前的位置是否已填充了该字符 if xx in []

if result[y] == lst[x]:

ok = False

break

if ok:

result[i] = lst[x]

permutation(i+1) # 填充下一个位置

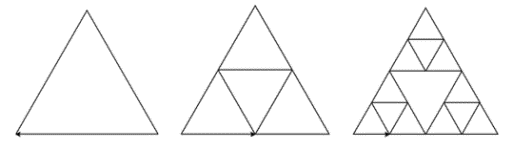
permutation(0)

4. 奇异三角形（非OJ题目）：

一个边长为x的0阶奇异三角形，是一个边长为x的等边三角形。

一个边长为x的n阶奇异三角形，是一个边长为x的等边三角形，三个角上分别是一个边长为x/2的n-1阶奇异三角形。

下图从左到右分别是0阶、1阶和2阶奇异三角形：



输入整数 n(0<=n<=5),绘制n阶奇异三角形。

提示：turtle.left(x)可以向左拐x度；turtle.right(x)可以向右拐x度；pos=turtle.pos()可以获得画笔当前位置，以后turtle.goto(pos)就可以移动画笔到那个位置；turtle.seth(x)可以设置画笔方向为角度x；绘图完成后调用turtle.done()可以保持绘图窗口。以上提示内容并非一定要用到。

解答：

import turtle

def triangle(size,n): #size是长度，n是阶数目，以画笔当前位置作为左下角，画一个边长为size，阶为n的奇异三角形

if n == 0: #0阶则直接画一个三角形，以笔的当前位置作为左下角

turtle.pendown()

turtle.seth(60) #设置方向为60度

turtle.fd(size) #前进size像素

turtle.right(120) #右转120度

turtle.fd(size)

turtle.right(120)

turtle.fd(size)

turtle.penup()

else:

pos = turtle.pos()

triangle(size/2,n-1)

turtle.goto(pos)

turtle.seth(60)

turtle.fd(size/2)

triangle(size/2,n-1)

turtle.goto(pos)

turtle.seth(0)

turtle.fd(size/2)

triangle(size/2,n-1)

turtle.setup(800,800) #窗口缺省位于屏幕正中间

turtle.speed(-1000)

turtle.penup()

turtle.goto(-200,-200)

turtle.pensize(2)

level =4

triangle(300,level)

turtle.done() #保持绘图窗口