1. 石头剪刀布(P071)：已知两人有不同的周期性出拳序列，比如一人总是“石头-布-石头-剪刀-石头-布-石头-剪刀……”循环。问出拳N次后，谁赢得多。

解答：

def Win(a,b):

if a == b:

return 0

if a == 0:

if b == 2:

return 1

else:

return -1

elif a == 2:

if b == 0:

return -1

else:

return 1

elif a == 5:

if b == 0:

return 1

else:

return -1

lst = input().split()

n ,na,nb = int(lst[0]),int(lst[1]),int(lst[2])

a = tuple(map(int,input().split()))

b = tuple(map(int,input().split()))

awin = 0

pa = 0

pb = 0

for i in range(n):

awin += Win(a[pa],b[pb])

pa = ( pa + 1 ) % na

pb = ( pb + 1 ) % nb

if awin == 0:

print("draw")

elif awin > 0:

print("A")

else:

print("B")

2. 统计数字字符个数(P065)：输入一行字符，统计出其中数字字符的个数。

解答：

s = input()

total = 0

for x in s:

if x >= "0" and x <= "9":

total += 1

print(total)

3. 大小写字母互换(P066)：把一个字符串中所有出现的大写字母都替换成小写字母，同时把小写字母替换成大写字母。

解答：

s = input()

for c in s:

if 'a' <= c <= 'z':

print(chr(ord(c) - 32 ),end="")

elif 'A' <= c <= 'Z':

print(chr(ord(c) + 32),end="")

else:

print(c,end="")

4. 过滤多余的空格(P064)：一个句子中也许有多个连续空格，过滤掉多余的空格，只留下一个空格。

解答：

s = input().split()

result = ""

for x in s:

result += x + " "

print(result.strip())

5. 找第一个只出现一次的字符(P067)：给定一个只包含小写字母的字符串，请你找到第一个仅出现一次的字符。

解答：

解法1：

s = input()

for i in range(len(s)):

if s[i] not in s[i+1:] and s[i] not in s[:i]:

print(s[i])

exit() #exit结束整个程序

print("no")

解法2：

s=input()

for c in s:

if s.count(c)==1:

print(c)

break

else:print("no")

解法3：最快，比前两种快n倍，n是字符串字符数目

xxx=list(input())

lst=[0]\*26 #lst[i]表示编码为ord('a')+i的字符的出现次数

flag=1

for i in xxx:

lst[ord(i) - ord('a')] +=1

for i in xxx:

if lst[ord(i)-ord('a')]==1:

print(i)

exit()

print("no")

6. 判断字符串是否为回文(P068):输入一个字符串，输出该字符串是否是回文串。回文串是指顺读和倒读都一样的字符串。比如：abba，cccdeedccc都是回文字符串。

解答：

s = input()

if s == s[::-1]:

print("yes")

else:

print("no")

7. 字符串最大跨距(P069)：有三个字符串S,S1,S2，想检测S1和S2是否同时在S中出现，且S1位于S2的左边，并S1和S2不重叠。计算满足上述条件的最右边的S2的起始点与最左边的S1的终止点之间的字符数目。

解答：

s,s1,s2 = input().split(",")

pos1 = s.find(s1)

pos2 = s.rfind(s2)

if pos1 == -1 or pos2 == -1:

print(-1)

exit()

if pos2 - pos1 >= len(s1):

print(pos2-(pos1+len(s1)))

else:

print(-1)

8. 找出全部子串位置(P070)：两个字符串s1,s2，找出s2在s1中所有出现的位置。

解答：

n = int(input())

for i in range(n):

s1,s2 = input().split()

found = False

start = 0

i = s1.find(s2,start)

while i != -1:

found = True

print(i,end = " ")

start = i + len(s2)

i = s1.find(s2, start)

if not found:

print("no")

else:

print("")

9. 万年历(P073)：给定年月日，求星期几。已知2020年11月18日是星期三。本题不但有公元前年份，还有公元0年，这个和真实的纪年不一样。

解答：

﻿def isLeapYear(n):

if n%4 ==0 and n%100!= 0 or n%400 == 0: #闰年

return True

return False

def yearDays(n):

if isLeapYear(n):

return 366;

return 365

monthDays = [-1,31,28,31,30,31,30,31,31,30,31,30,31]

def countWeekDay(year,month,date):

days = 0 #从2012-01-22开始过了多少天

if year >= 2012:

for y in range(2012,year):

days += yearDays(y)

if isLeapYear(year):

monthDays[2] = 29

else:

monthDays[2] = 28

for i in range(1,month):

days += monthDays[i]

days += date

days -= 22 #2012年1月22日是星期天

if days >= 0:

return days % 7

else:

return (7 + days) % 7

else:

for y in range(2011,year,-1):

days += yearDays(y)

if isLeapYear(year):

monthDays[2] = 29

else:

monthDays[2] = 28

for i in range(12,month,-1):

days+= monthDays[i]

days+= monthDays[month] - date

days += 22

return (7 - (days % 7)) % 7

weekDays = ("Sunday","Monday","Tuesday","Wednesday","Thursday", "Friday","Saturday")

n = int(input())

for i in range(n):

lst = input().split()

year,month,date = int(lst[0]),int(lst[1]),int(lst[2])

if isLeapYear(year):

monthDays[2] = 29

else:

monthDays[2] = 28

if month <1 or month > 12 or date < 1 or date > monthDays[month]:

print("Illegal")

else:

print(weekDays[countWeekDay(year,month,date)])

1. 成绩排序(P093)：给出一些学生的姓名和成绩，将学生按成绩从高到低排序。成绩相同的学生，按照姓名从小到大排序。

解答：

n = int(input())

students = []

for i in range(n):

s = input().split()

students.append((s[0],int(s[1])))

students.sort(key = lambda x : (-x[1],x[0]))

for x in students:

print(x[0],x[1])

2. 图像模糊处理 (P061)：给定n行m列的图像各像素点的灰度值，要求用如下方法对其进行模糊化处理：

（1）四周最外侧的像素点灰度值不变；

（2）中间各像素点新灰度值为该像素点及其上下左右相邻四个像素点原灰度值的平均（舍入到最接近的整数）。

解答：

n,m = map(int,input().split())

a = []

b = []

for i in range(n):

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

a.append(lst)

b.append(lst.copy())

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

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

b[i][j] = round((a[i][j] + a[i-1][j] +

a[i+1][j] + a[i][j-1] + a[i][j+1])/5)

for i in range(0,n):

for j in range(0,m):

print(b[i][j],end = " ")

print("")

3. 向量点积计算(P072)：给定两个n维向量**a**=(a1,a2,...,an)和**b**=(b1,b2,...,bn)，求点积**a**·**b**=a1b1+a2b2+...+anbn。

解答：

n = int(input())

a = tuple(map(int,input().split()))

b = tuple(map(int,input().split()))

total = 0

for i in range(n):

total += a[i] \*b[i]

print(total)

4. 病人排队(P075)：请将登记的病人按照以下原则排出看病的先后顺序：1)老年人（年龄 >= 60岁）比非老年人优先看病。2) 老年人按年龄从大到小的顺序看病，年龄相同的按登记的先后顺序排序。3)非老年人按登记的先后顺序看病。

解答：

def f(x):

if x[1] >= 60:

return (0,-x[1],x[2])

else:

return (1,0,x[2])

n = int(input())

p = []

for i in range(n):

s = input().split()

p.append((s[0],int(s[1]),i))

p.sort(key = f)

for i in range(n):

print(p[i][0])

5. 矩阵乘法(P077)：给定两个矩阵，计算其乘积。

解答：

n,m,k = map(int,input().split())

a = []

b = []

for i in range(n):

a.append(list(map(int,input().split())))

#a的每个元素都是一个列表，那么a就是一个二维列表，即矩阵

#a的每个元素相当于矩阵的一行

for i in range(m):

b.append(list(map(int,input().split())))

c = [[0 for j in range(k)] for i in range(n)]

#c是一个 3\*3矩阵，所有元素都是0

for i in range(n):

for j in range(k): #计算c[i][j]

for kk in range(m):

c[i][j] += a[i][kk] \* b[kk][j]

for i in range(n):

for j in range(k):

print(c[i][j],end = " ") #行末多空格没关系

print("")

6. 回文子串(P076)：给定一个字符串，输出所有长度至少为2的回文子串。子串长度小的优先输出，若长度相等，则出现位置靠左的优先输出。

解答：

def isHw(s):

return s == s[::-1]

s = input()

L = len(s)

result = []

for i in range(L):

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

tmps = s[i:j+1]

if isHw(tmps):

result.append((j-i,i,tmps))

result.sort()

for x in result:

print(x[2])

7. 校园食宿预订系统(P081)：某校园为方便学生订餐，推出食堂预定系统。食宿平台会在前一天提供菜单，学生在开饭时间前可订餐。 食堂每天会推出m个菜，每个菜有固定的菜价和总份数，售卖份数不能超过总份数。 假设共有n个学生点餐，每个学生固定点3个菜，当点的菜售罄时, 学生就买不到这个菜了。 请根据学生预定记录，给出食堂总的预定收入。

解答：

n,m = map(int,input().split())

dishes = {}

for i in range(m):

s = input().split()

dishes[s[0]] = [int(s[1]),int(s[2])]

total = 0

for i in range(n):

s = input().split()

for d in s:

if dishes[d][1] > 0:

dishes[d][1] -= 1

total += dishes[d][0]

print(total)