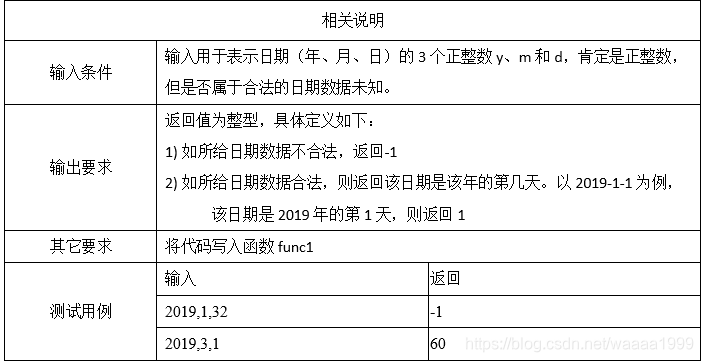
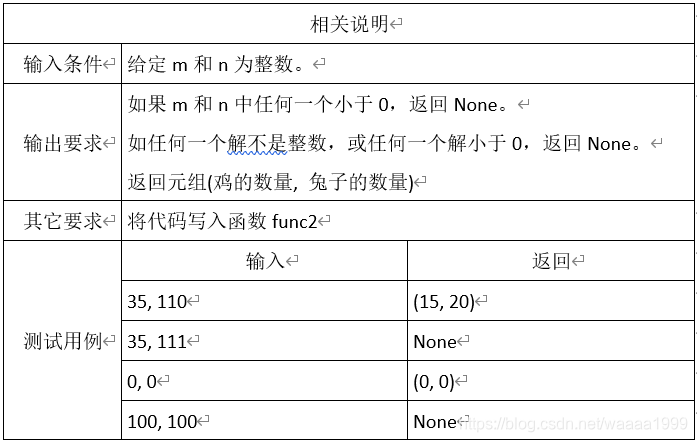
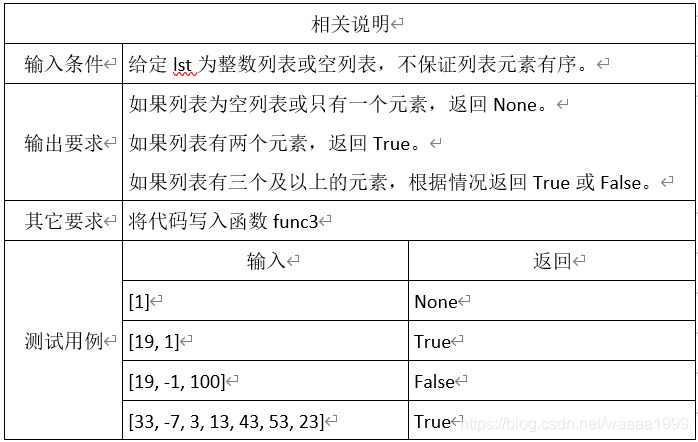
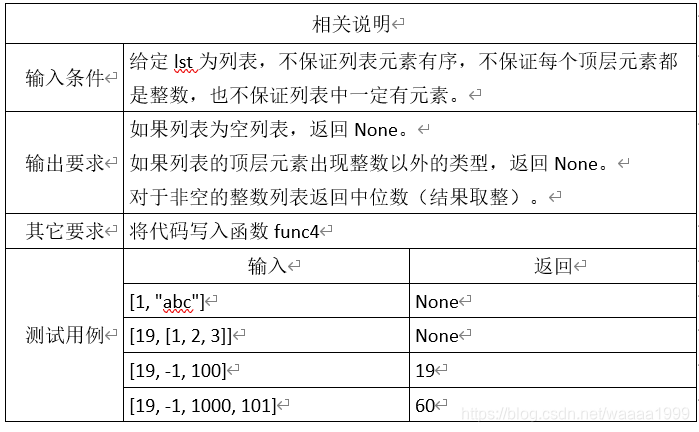
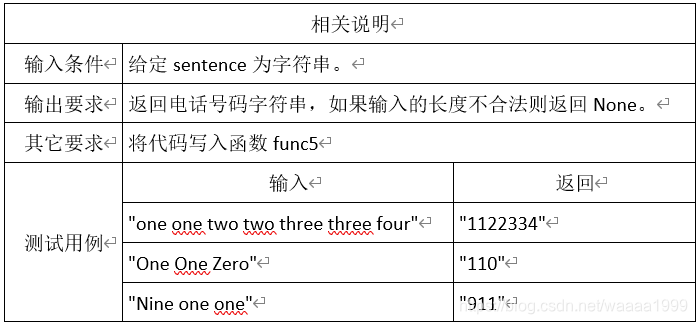
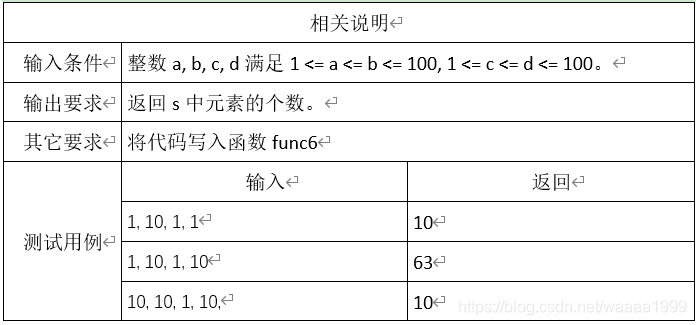
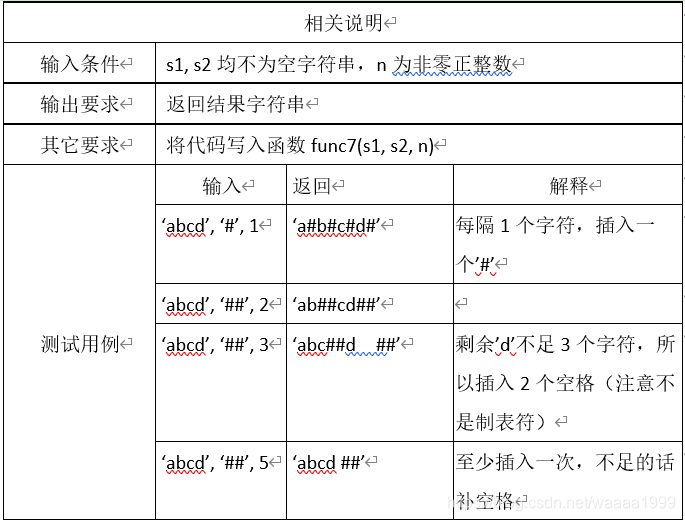
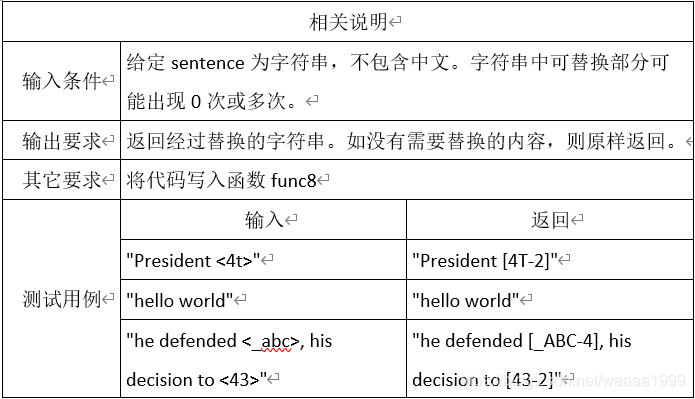
**试题**

1. 给定用于表示年、月、日的3个整数y、m和d，判定该日期是该年的第几天。  
   
2. 已知每只鸡有2只脚，每只兔子有4只脚，求解鸡兔同笼问题。给定鸡和兔子的头的总数m和脚的总数n。求鸡和兔子的数量。  
   
3. 判定一个整数列表里的元素排序后能否构成等差数列。  
   
4. 计算并返回整数列表的中位数。中位数的含义：对于一个有序的数列，排列位置位于整个列表中间的那个元素的值即为中位数。如果数列有偶数个值，取最中间的两个数值的平均数作为中位数。  
   
5. 输入一个包含3到11个单词的字符串，单词与单词之间用一个空格分开，最前和最后没有空格，其中的单词一定是0-9数字的英文单词（单词的字母可能大写也可能小写）。请编写程序将其转换为阿拉伯数字的字符串。  
   
6. 给定4个整数a, b, c, d，求集合s = {x / y | a <= x <= b, c <= y <= d}中元素的个数。  
   
7. 输入两个字符串参数s1, s2（均不为空字符串），和一个非零正整数n。请按照如下规则将字符串s2插入到s1中，并返回生成的字符串：  
   （1）s1中每隔n个字符，插入一次s2  
   （2）如果最后一次不足n个字符，则先用空格符号补全到n个字符，然后插入一次s2  
   
8. 给定一个字符串，找出其中“”形式的标签片段，并替换成“[TAG-len]”形式的片段。其中tag是由字母、数字和下划线构成可变的标签文本，TAG是将tag中的英文字母全部转换成大写字母后的形式，len是TAG的长度，详见测试用例。  
   

**参考答案1**

import re

def func1(y,m,d):

if m>12 or d>31:

return -1

if y%4==0:

rlb=[31,29,31,30,31,30,31,31,30,31,30,31]

if d>rlb[m-1]:

return -1

else:

t=0

for i in range(m-1):

t+=rlb[i]

t+=d

return t

else:

rlb=[31,28,31,30,31,30,31,31,30,31,30,31]

if d>rlb[m-1]:

return -1

else:

t=0

for i in range(m-1):

t+=rlb[i]

t+=d

return t

def func2(m,n):

if m<0 or n<0:

return None

x=(4\*m-n)/2

y=(n-2\*m)/2

if x<0 or y<0 or x!=int(x) or y!=int(y):

return None

else:

return (int(x),int(y))

def func3(lst):

if len(lst)<=1:

return None

elif len(lst)==2:

return True

else:

lst.sort()

d=lst[1]-lst[0]

for i in range(len(lst)-1):

if lst[i+1]-lst[i]!=d:

return False

return True

def func4(lst):

l=len(lst)

if l==0:

return None

for i in lst:

if isinstance(i,int)==False:*#参数先对象后类型*

return None

lst.sort()

if l%2==0:

zws=int((lst[int(l/2)]+lst[int(l/2-1)])/2)

else:

zws=lst[int((l-1)/2)]

return zws

def func5(s):

s=s.lower()*#string.lower()有返回值*

s=s.split()

if len(s)<3 or len(s)>11:

return None

ss=''

d={'zero':'0','one':'1','two':'2','three':'3','four':'4',

'five':'5','six':'6','seven':'7','eight':'8','nine':'9'}

for i in s:

ss+=d[i]

return ss

def func6(a,b,c,d):

s=[]

for i in range(a,b+1):

for j in range(c,d+1):

s.append(i/j)

s=set(s)

return len(s)

def func7(s1,s2,n):

s=[]

count=n

ss=''

i=0

for i in range(int(len(s1)/n)):

s.append(s1[i\*n:i\*n+n])

if i\*n+n<len(s1):

m=s1[i\*n+n:]

while len(m)<n:

m+=' '

s.append(m)

if len(s1)<n:

m=s1

while len(m)<n:

m+=' '

s.append(m)

for i in s:

ss+=i+s2

return ss

def func8(s):

tag=re.findall('<[a-zA-Z0-9\_]\*>',s)

tagl=''

taglen=[]

for i in tag:

tagl='['+i.upper()[1:-1]+'-'+str(len(i)-2)+']'

taglen.append(tagl)

tagl=''

for i in taglen:

s=re.sub('<[a-zA-Z0-9\_]\*>',i,s,1)

return s

if \_\_name\_\_=='\_\_main\_\_':

pass

'''

print(func1(2019,1,32))

print(func1(2019,3,1))

print(func2(35,110))

print(func2(35,111))

print(func2(0,0))

print(func2(100,100))

print(func3([1]))

print(func3([19,1]))

print(func3([19,-1,100]))

print(func3([33,-7,3,13,43,53,23]))

print(func4([1,'abc']))

print(func4([19,[1,2,3]]))

print(func4([19,-1,100]))

print(func4([19,-1,1000,101]))

print(func5('one one two two three three four'))

print(func5('One One Zero'))

print(func5('Nine one one'))

print(func6(1,10,1,1))

print(func6(1,10,1,10))

print(func6(10,10,1,10))

print(func7('abcd','#',1))

print(func7('abcd','##',2))

print(func7('abcd','##',3))

print(func7('abcd','##',5))

print(func8('President <4t>'))

print(func8('hello world'))

print(func8('he defended <\_abc>, his decision to <43>'))

'''

**参考答案2**

def func1(y, m, d):

if y % 400 == 0 or (y % 4 == 0 and y % 100 != 0):

days\_feb = 29

else:

days\_feb = 28

md = [31, days\_feb, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

if m > 12 or d > md[m - 1]:

return -1

sum = 0

for i in range(m - 1):

sum += md[i]

sum = sum + d

return sum

def func2(m, n):

if m < 0 or n < 0:

return None

x, y = (4\*m - n)/2, (n-2\*m)/2

x\_int, y\_int = int(x), int(y)

if x < 0 or y < 0 or x\_int!=x or y\_int!=y:

return None

return x\_int, y\_int

def func3(lst):

n = len(lst)

if n == 0 or n == 1:

return None

if n == 2:

return True

lst.sort()

dif = [lst[x + 1] - lst[x] for x in range(len(lst) - 1)]

for x in range(1, len(dif)):

if dif[x - 1] != dif[x]:

return False

return True

def func4(lst):

n = len(lst)

if n == 0:

return None

for item in lst:

if type(item) != int:

return None

lst.sort()

if n % 2 == 0:

return (int((lst[n // 2 - 1] + lst[(n // 2)]) // 2))

else:

return (int(lst[n // 2]))

def func5(sentence):

dict1 = {'one': '1', 'two': '2', "three": "3", "four": "4", "five": "5",

"six": "6", "seven": "7", "eight": "8", "nine": "9", "zero": "0"}

words = sentence.split()

if len(words) < 3 or len(words) > 11:

return None

else:

res = ""

for item in words:

res = res + dict1[item.lower()]

return res

def func6(a, b, c, d):

s = {p / q for p in range(a, b + 1) for q in range(c, d + 1)}

return len(s)

def func7(s1, s2, n):

tokens = [s1[i: i+n] for i in range(0, len(s1), n)]

tokens[-1] += ' ' \* (n - len(tokens[-1]))

return s2.join(tokens) + s2

*# encoding=utf-8*

import re

def rep(m):

s = m.group() *# 返回一个或多个子模式内容*

s = s.upper()

return "[" + s[1: len(s)-1] + "-" + str(len(s) - 2) + "]"

def func8(sentence):

pattern = re.compile("<\w+>") *# s = pattern.findall(sentence)*

sentence = pattern.sub(rep, sentence)

return sentence

if \_\_name\_\_ == "\_\_main\_\_":

print(func1(1991, 1, 1))

print(func1(2020, 2, 29))

print(func1(2019, 2, 29))

print(func1(2015, 3, 13))

print(func2(35, 110))

print(func2(35, 111))

print(func2(0, 0))

print(func2(100, 100))

print(func3([1]))

print(func3([19,1]))

print(func3([19,-1,100]))

print(func3([33, -7, 3, 13, 43, 53, 23]))

print(func4([1, "abc"]))

print(func4([19, [1, 2, 3]]))

print(func4([19,-1,100]))

print(func4([19, -1, 100]))

print(func5("one one two two three three four"))

print(func5("One One Zero"))

print(func5("Nine one one"))

print(func6(1, 10, 1, 1))

print(func6(1, 10, 1, 10))

print(func6(10, 10, 1, 10))

print(func7('abcd', '#', 1))

print(func7('abcd', '##', 2))

print(func7('abcd', '##', 3))

print(func7('abcd', '##', 5))

*# print(func8("President <4t>"))*

*# print(func8("hello world"))*

print(func8("he defended <\_abc>, his to <43>"))

**其它**

**考试说明**

1. 考试时间：合计180分钟，选择题部分30分钟交卷，允许提前交卷。
2. 考试过程中，不能连接未经指定网站或服务器。
3. 闭卷考试部分，不能查阅任何类型的参考资料。
4. 开卷考试部分，可以查阅纸质文档，不能查阅除Python编程环境自带帮助文件以外的任何类型的电子文档。
5. 考试过程中，不得使用任何形式的电子存储设备，不可使用手机。
6. 违反上述2-5条者，视为考试作弊。

**选择题答题方式（25分，闭卷，严禁使用python编程环境进行尝试）**  
8. 打开浏览器，在地址栏中输入http://192.168.125.3，点击相应链接进入登录页面。  
9. 按要求输入两遍自己的学号。  
10. 点击“登录”按钮即可进入答题页面。如考试尚未开始，系统会进入等待页面并倒计时。考试开始时间到，系统会自动进入答题页面。  
11. 在页面左侧选择题号，页面右侧即会显示相应的题目。考生只需点击选择相应的选项。  
12. 答题过程中如关闭浏览器或出现系统故障导致计算机重新启动，系统不会丢失之前已经完成的题目的答案。考生可以打开浏览器重新登录并继续考试。  
13. 答题完成后，点击“交卷”按钮即可完成交卷。交卷后不能再次登录系统继续考试。  
14. 考试结束时间到，系统会自动收卷。  
**编程题注意事项与提交方式（75分，开卷）**  
15. Sample.py中的函数名、参数数量和顺序不可以修改。  
16. 调用自己写的函数、自己测试的代码等，请写入到if **name**==“**main**”:，不要写到全局环境中。  
17. 不要在全局环境中调用input()。  
18. 每个函数中不需要print,而是用return返回结果。  
19. 不要使用关键字（if、else、 break、 def等，会导致语法错误）作为自己的变量名，也不要使用内嵌名字（如list、 int、 input等，没有语法错误，但是很危险）作为自己的变量名，例如：list = []  
20. 代码中不要出现任何中文。  
21. 确保提交的时候，自己的程序可以正常运行，不要遗留任何语法错误。  
22. 所有脚本程序内容必须仅包含在一个脚本程序文件（py文件）中。  
23. 提交前务必关闭IDLE或PyCharm编程环境。  
24. 在浏览器的地址栏中输入http://192.168.125.3，点击相应链接进入提交页面。  
25. 按要求输入两遍自己的学号。  
26. 点击“选择文件”按钮，选择自己的脚本程序文件。点击“提交”按钮提交。  
27. 如提交成功，系统会显示相关信息。如果提交不成功，请重复步骤16-18。  
28. 提交成功后，可点击“查看内容”按钮检查提交的内容。  
29. 请根据图所示的说明严格规范源文件结构。