1. 找出所有整数（P084）：给一段文字，可能有中文，把里面的所有非负整数都找出来，不需要去掉前导0。如果碰到 "012.34"这样的就应该找出两个整数 012和34,碰到 0.050,就找出 0和050。

解答：

import re

#your code starts here

m = r"\d+"

#your code ends here

while True:

try:

s = input()

lst = re.findall(m,s)

for x in lst:

print(x)

except:

break

2. 找出所有整数和小数（P085）：给一段文字，可能有中文，把里面的所有非负整数和小数找出来，不需要去掉前导0或小数点后面多余的0， 然后依次输出。

解答：

import re

import sys

#your code starts here

m = r"\d+\.\d+|\d+"

#your code ends here

while True:

try:

s = input()

lst = re.findall(m,s)

for x in lst:

print(x)

except:

break

3. 找出小于100的整数（P086）：有给定的两行输入，在每一行的输入中提取在[0,100)内的整数(不包括100)并依次输出。注意要排除负数。

解答：

import re

m = r"(^|[^0-9-])(\d{1,2})([^0-9]|$)"

for i in range(2):

s = input()

lst = re.findall(m,s)

for x in lst:

#your code ends here

print(x[1])

#your code starts here

4. 有一个文本文件包含所有学生的学号和姓名，还有一个文本文件包含某次openjudge作业学生的提交排名记录。排名记录中包含学生的昵称和通过的题目数。昵称可能包含姓名，可能包含学号，也可能都包含。给定一个算分规则，请根据这两个文件生成一个学生分数的文件。本题详情及用到的两个文本文件请见附件资源包。

解答：

import sys

f = open("id.txt","r",encoding="utf-8")

students = f.readlines()

f.close()

students.sort()

f = open("finalscore.txt","r",encoding="utf-8")

scores = f.readlines()

f.close()

f = open(sys.argv[1],"w",encoding="utf-8")

for st in students:

st = st.strip()

if st == "":

continue;

id,name = st.split("\t")

found= False

for sc in scores:

lst = sc.split("\t")

if id in lst[1] or name in lst[1]:

found = True

if lst[2].isdigit():

num = int(lst[2])

else:

num = int(lst[3])

# 分数计算办法: 1题50,2题60,此后每做一题加4分

finalscore = 0

if num == 0:

finalscore = 0

elif num == 1:

finalscore = 50

elif num == 2:

finalscore = 60

else:

finalscore = 60+(num-2)\*4

f.write("%s\t%s\t%d\t%d\n" % (id,name,num,

finalscore))

#print(id,name,num ,finalscore)

break

if not found:

f.write("%s\t%s\t%d\t%d\n" % (id,name,0,0))

f.close()

1. 密码判断（P087）：用户密码的格式是： 1） 以大写或小写字母开头 2) 至少要有8个字符，最长不限 3) 由字母、数字、下划线或 '-' 组成 输入若干字符串，判断是不是符合密码的条件。如果是，输出 yes 如果不是，输出 no。

解答：

import re

#your code starts here

m = r"[a-zA-Z][a-zA-Z\d\_-]{7}[a-zA-Z\d\_-]\*$"

#your code ends here

while True:

try:

s = input()

if re.match(m,s) != None:

print("yes")

else:

print("no")

except:

break

2. 找<>中的数（P088）：输入一串字符，将输入中的，在<>里面的，没有前导0的少于4位的整数依次输出。单独的0也要输出。

解答：

import re

pt = '<(0|[1-9][0-9]{0,2})>'

#在<>里面的，没有前导0的少于4位的整数依次输出。单独的0也要输出。

n = int(input())

for i in range(n):

str = input()

y = re.findall(pt,str)

if y == []:

print("NONE")

else:

for x in y:

print(x,end = " ")

print("")

3. 电话号码（P090）："<X>和它右边离他最近的"</X>"构成一个tag。电话号码由区号和号码构成，行如(20)-784。请找出tag中的电话号码。

解答：

import re

pt1 = "(<([a-z]+)>)(.\*?)(</\\2>)"

pt2 = "(\((\d{1,2})\))-(\d+)"

n = int(input())

for ii in range(n):

s = input()

found = False

for x in re.finditer(pt1,s):

head = x.group(1)

tail = x.group(4)

str = x.group(3)

result = []

for y in re.finditer(pt2,str):

if len(y.group(3)) == 3:

result.append(y.group(2))

if len(result) > 0:

found = True

print(head,end="")

for i in range(len(result)):

if( i == 0):

print(result[i],end="")

else:

print("," + result[i],end="")

print(tail)

if found == False:

print("NONE")

4. 编写一个能在指定文件夹的指定后缀名的文本文件中寻找指定字符串的程序findtext.py，用法如下：

python findtext.py 文件夹 要找的字符串 后缀名1 后缀名2 后缀名3......

用法示例：

python findtext.py "c:\tmp\files" "XXXXX" txt html csv

程序在c:\tmp\files文件夹下的所有后缀名为".txt"或".html"或".csv"的文件中寻找字符串"XXXXX"，还要递归查找所有子文件夹、子子文件夹下的文件，如果找到哪个文件包含" XXXXX"，则输出该文件带路径的文件名。注意，由于这些文件编码可能为utf-8，也可能为gbk，要求都能处理。对实在处理不了的文件，就跳过它，不能让程序RE中止。

解答：

import os

import sys

def walkPath(path,strToFind,exts):

lst = os.listdir(path)

for x in lst:

actualFileName = path + "/" + x # x不包括路径

if os.path.isfile(actualFileName):

ok = False

for ext in exts:

if x.lower().endswith(ext):

ok = True

break

if ok:

f = open(actualFileName,"r",encoding="utf-8")

while True:

line = None

try:

line = f.readline()

except:

pass

if line == "":

break

if strToFind in line:

print("found in : ",actualFileName)

break

f.close()

else:

if walkPath(actualFileName,strToFind,exts ):

return True

return False

exts = []

for x in sys.argv[3:]:

exts.append("." + x)

walkPath(sys.argv[1],sys.argv[2],exts)