装 订 线

考试时间

1

班级 姓名 学号 教学班名称 序号 任课教师 -

密 封 线

**广东财经大学考试答题纸（格式一）**

课程

注意：答题请注明题号，有选择题或判断题的先答，其余的按顺序排列。

（

）单项选择题

（

）多项选择题

（

）判断题

|  |  |  |  |
| --- | --- | --- | --- |
| 答案 | 题号 | 答案 | 题号 |
|  | 11 |  | 1 |
|  | 12 |  | 2 |
|  | 13 |  | 3 |
|  | 14 |  | 4 |
|  | 15 |  | 5 |
|  | 16 |  | 6 |
|  | 17 |  | 7 |
|  | 18 |  | 8 |
|  | 19 |  | 9 |
|  | 20 |  | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| 答案 | 题号 | 答案 | 题号 |
|  | 11 |  | 1 |
|  | 12 |  | 2 |
|  | 13 |  | 3 |
|  | 14 |  | 4 |
|  | 15 |  | 5 |
|  | 16 |  | 6 |
|  | 17 |  | 7 |
|  | 18 |  | 8 |
|  | 19 |  | 9 |
|  | 20 |  | 10 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 答案 | 题号 | 答案 | 题号 | 答案 | 题号 |
|  | 31 |  | 11 |  | 1 |
|  | 32 |  | 12 |  | 2 |
|  | 33 |  | 13 |  | 3 |
|  | 34 |  | 14 |  | 4 |
|  | 35 |  | 15 |  | 5 |
|  | 36 |  | 16 |  | 6 |
|  | 37 |  | 17 |  | 7 |
|  | 38 |  | 18 |  | 8 |
|  | 39 |  | 19 |  | 9 |
|  | 30 |  | 20 |  | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| 复核人 | 阅卷人 | 得分 | 题号 |
|  |  |  | 一 |
|  |  |  | 二 |
|  |  |  | 三 |
|  |  |  | 四 |
|  |  |  | 五 |
|  |  |  | 六 |
|  |  |  | 七 |
|  |  |  | 八 |
|  |  |  | 总分 |

一

import math

# 调用阶乘函数，计算并输出用户输入的20以内的正整数的阶乘。提示信息为：请输入一个20以内的正整数

print(math.factorial(int(input("请输入一个20以内的正整数: "))))

import random,string

# 调用random.choices()方法，随机从ascii字母和数字串中取5个字符生成列表，拼接列表元素为字符串输出

print(''.join(random.choices(string.ascii\_letters + string.digits, k=5)))

import os

# 在当前文件夹下创建名为datatxt的文件夹

if not os.path.exists('datatxt'): os.mkdir('datatxt')

set1={'hour':'23','miniute':56,'second':'37'}

# 以':'为分隔符，调用join()方法和map方法输出字典set1的值，输出结果为：23:56:37

print(':'.join(list(map(str, set1.values()))))

student='我的名字叫{},是广东财经大学的学生'.format(input('请输入你的姓名：'))

# 用逆序索引从student切片输出你的姓名,其中姓名输入自己的名字

print(student[-student.rfind('}'):])

lst,tu=[33,44,55,66],(77,88,99)

# 把元组tu解包后添加到列表lst

print(lst.extend(list(tu)))

my\_lst=['73','13','9','82','66']

# 对列表my\_lst按整数排序，降序输出

print([int(i) for i in my\_lst].sort(reverse=True) )

set2={-4,-2,8,4,-6,8,6}

# 取集合set2中大于零的元素并求平方，生成列表输出

print([i\*\*2 for i in [i for i in list(set2) if i > 0]])

class\_py,student\_no=['审计学2班','审计学2班','行政管理1班'],(56,62,55,77)

# 组合class\_py和student\_no，生成二元元组合列表并输出，输出结果为：[('审计学2班', 56), ('审计学2班', 62), ('行政管理1班', 55)]

print(list(zip(class\_py, student\_no)))

s=set(('Python','is','beautiful','?'))

# 用条件表达式实现：如果'?'是集合s元素，删除该元素，如果不是，输出'该元素不存在的信息'

s.discard('?') if '?' in s else print("该元素不存在")

s2019,s2018=['华为','三星','Apple','vivo','oppo'],('三星','Apple','华为','荣耀','wp')

# 用集合返回不同时属于列表s2019和元组s2018的元素，输出结果为：{'wp', 'vivo', 'oppo', '荣耀'}

print(set(s2019).symmetric\_difference(set(s2018)))

course={'python': 95, 'Java': 76, 'C语言': 82, '数据库原理': 88}

# 调用update()方法修改数据库原理的成绩为83

course.update({'数据库原理': 83});print(course)

s=' 静夜思\n床前明月光,疑是地上霜。\n举头望明月，低头思故乡。\n'

# 去掉首尾空白后将字符串s切分为列表并输出，输出结果为：['静夜思', '床前明月光,疑是地上霜。', '举头望明月，低头思故乡。']

print(s.strip().split('\n'))

with open('./data/poem.txt', 'w', encoding='utf-8') as f:

# 将字符串'江雪\n千山鸟飞绝，万径人踪灭。\n孤舟蓑笠翁，独钓寒江雪。\n'写入打开的文件

f.write('江雪\n千山鸟飞绝，万径人踪灭。\n孤舟蓑笠翁，独钓寒江雪。\n')

with open('./data/poem.txt', 'r', encoding='utf-8') as f:

# 读取文件的所有行，每一行去掉换行符后添加到列表data\_lst

print([line.strip() for line in f.readlines()])

import os

# 重命名peom.txt文件为江雪.txt并保存到datatxt文件夹下

if not os.path.exists('./datatxt/江雪.txt'):os.rename('./data/poem.txt', './datatxt/江雪.txt')

import numpy as np

# 从data文件夹下的’8.6 score.csv‘文件获取数据，返回字符串数组

print(np.genfromtxt('./data/8.6 score.csv', dtype=str,encoding="utf8"))

import pandas as pd

# 读取data文件夹下的’8.6 score.csv‘文件的数据，转换为二维列表输出,语句太长可加续行符

print(pd.read\_csv('./data/8.6 score.csv').values.tolist())

二

错误：NameError: name 'string' is not defined

改正：import string

错误：NameError: name '\_digit' is not defined

改正：letter = 0

digit = 0

chinese = 0

other = 0

错误：asdasd

1 0 0 0

2 0 0 0

3 0 0 0

4 0 0 0

5 0 0 0

6 0 0 0

改正：print放到开始

错误：没有统计其他字符的个数s

改正：加入

else:

other += 1

最终版：

import string

my\_string = input()

letter = 0

digit = 0

chinese = 0

other = 0

for c in my\_string:

if c in string.ascii\_letters:

letter=letter+1

elif c in string.digits:

digit+=1

elif c not in string.printable:

chinese+=1

else:

other += 1

print(letter, digit, chinese,other)

三

1、simple\_set.py

函数 fil\_lst 的作用是返回一个列表，这个列表包含符合以下条件的 n 位数：

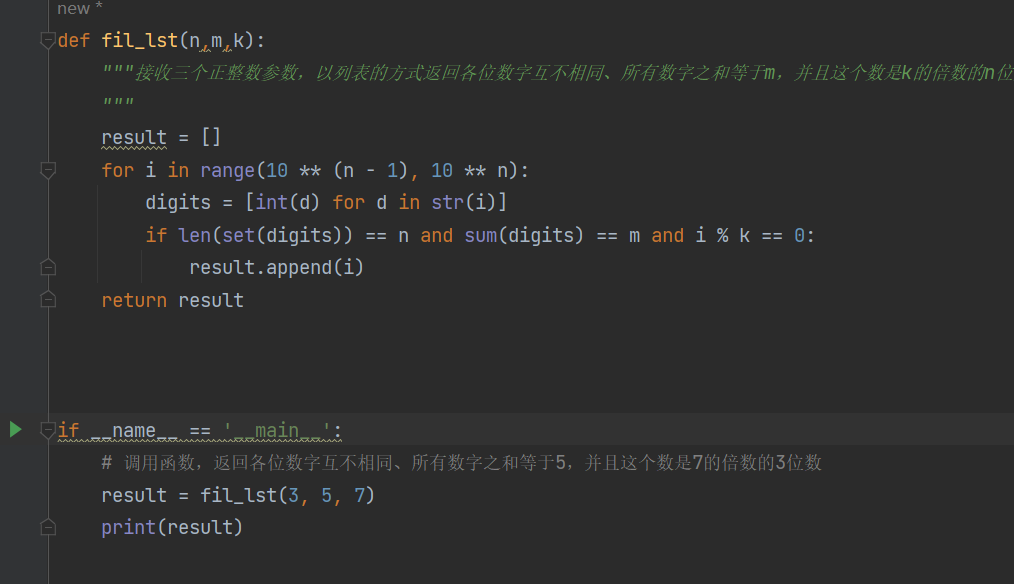
每一位上的数字互不相同

所有数字之和等于 m

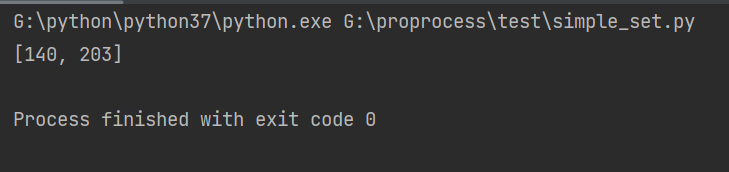
这个数是 k 的倍数

为了实现这个功能，代码使用了一个嵌套的 for 循环，在循环中遍历所有可能的数字组合。对于每个数字组合，将数字拆分为单个数字判断是否满足上述三个条件。满足就添加到列表。

完整代码：



测试用例：



2、sushu.py

代码思路如下：

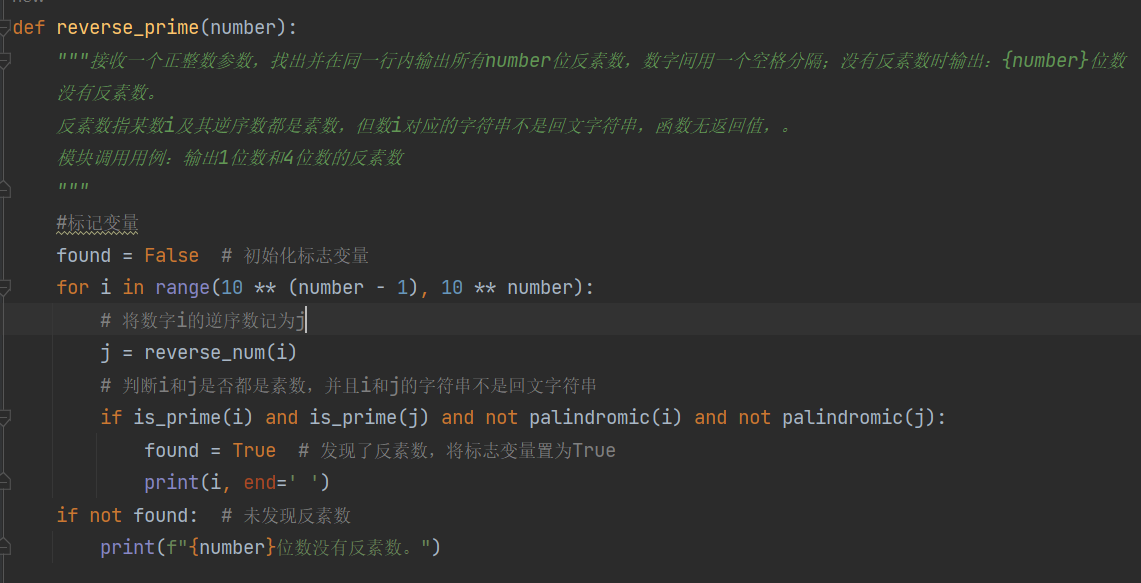
定义三个基础函数：is\_prime()用于判断数字是否是素数；palindromic()用于判断数字是否是回文数；reverse\_num()用于获取数字的逆序数。

定义函数reverse\_prime()，接收一个数字位数作为参数。

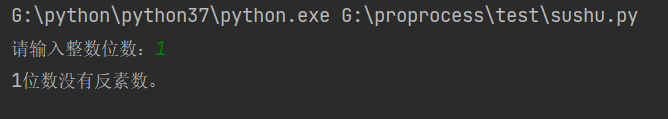
遍历所有符合位数的数字，判断其是否是反素数。如果是，则输出该数字。

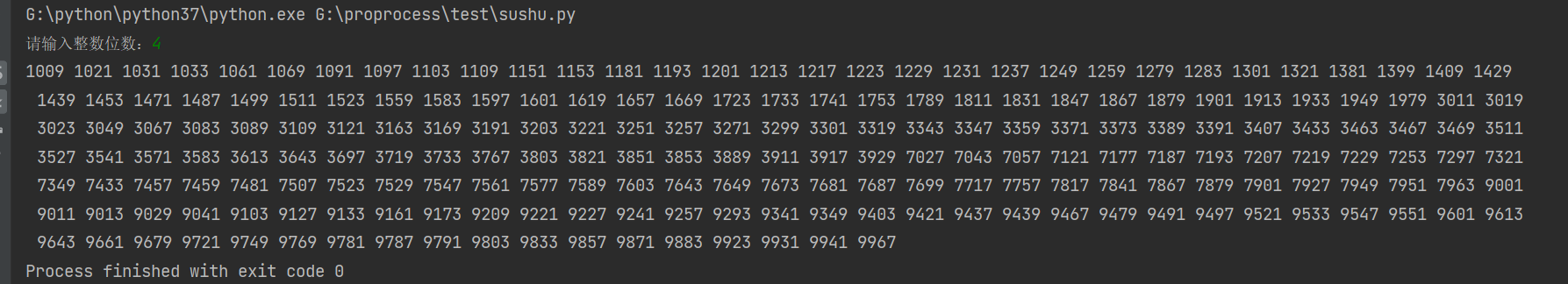
如果在遍历结束后没有发现反素数，则输出"{number}位数没有反素数。"的提示信息。

完整代码：



测试用例：





四

import pandas as pd # 导入pandas库起别名为pd

data=[]

print('请输入五门课程，用空格分隔')

data\_course=input('请输入课程：')

data.append(['姓名']+data\_course.split(' ')) # 用户输入切分为列表添加到创建的列表

print('请依次输入姓名和五门课程成绩，用空格分隔')

while True:

data\_score=input('请输入姓名和成绩数据：')

if data\_score=='':

break # 用户输入回车退出循环，中止输入

data.append(data\_score.split(' ')) # 用户输入切分为列表添加到创建的列表

with open('./data/myscore.csv','w',encoding='utf-8') as f:

for x in data:

f.write(','.join(map(str, x)) + '\n') # 列表的元素转换为字符串写入文件

file = './data/myscore.csv' # 指定要读取的文件

score = pd.read\_csv(file) # 读文件myscore.csv中的数据到dataframe对象中

scoreSum = [sum(x[1:]) for x in score.values.tolist()] # 计算总分

average\_scores = [round(sum(x[1:]) / (len(x) - 1), 2) for x in score.values.tolist()] # 计算每位同学的平均分,保留两位小数

scoreMean\_course=round(score.mean(numeric\_only=True),2) # 计算每门课程的平均分，保留两位小数

scoreMax\_course=score.max(numeric\_only=True) # 计算每门课程的最高分

scoreMedian\_student=round(score.median(axis=1,numeric\_only=True),2) # 计算每位同学成绩的中位数，保留两位小数

scoreMedian\_course= round(score.median(axis=0,numeric\_only=True),2) # 计算每门课程成绩的中位数，保留两位小数

print(dict(scoreMean\_course)) # 用字典的方式输出每门课程的平均分

print(dict(scoreMedian\_student)) # 用字典的方式输出每位同学成绩的中位数

print(dict(scoreMedian\_course)) # 用字典的方式输出每门课程成绩的中位数

print(dict(scoreMax\_course)) # 用字典的方式输出每门课程的最高分

score['总分'] = scoreSum # 在数据最后加上总分列

score['平均分'] = average\_scores # 在数据最后加上平均分列

score = score.sort\_values(by=['总分'], ascending=False) # 总分降序排序

score.to\_csv(file) # 写回到文件

print(score.sort\_values('总分', ascending=False)[:3]) # 降序输出总分排前三的记录

print(score.sort\_values('平均分', ascending=True)[-3:]) # 升序输出平均分排后三的记录

结果示例：

