**一、填空题**

**1.**在Python中，执行循环结构时，for i in range(5) ，i的取值是 ( 0,1,2,3,4) 。

**2.**执行以下Python代码后，变量result的值是 fox 。

**s = "The quick brown fox"**

**result = s.split()[-1]**

**3.**执行下面代码的结果是 A python program 。

**s = "****a python program"**

**result =s.title().replace("P", "p")**

**print(result)**

**4.**某同学编写一段程序代码如下，在运行时出错，错误提示为：“NameError: name 'num2' is not defined”

**1：num1 = 10**

**2：****result = num1 + (num2)**

**3：print(result)**

如果这位同学不想写这行num2=20语句，他应该怎么更改代码才能保证程序运行打印出来的结果是30，他要更改的代码行号和该行更改的具体内容为： result = num1 + (num2: =20)

**5.**某同学想设计一个通讯录系统，需要存储一个学生信息如下：姓名：李华，电话：13804512345，微信：Lihua。他应该选择 字典 数据类型进行学生信息存储。

**6.**以下代码片段中，data 是一个包含嵌套字典和列表的复杂数据结构。请补全代码，以便访问键 'name' 对应的值，该值存储在字典中，而该字典又位于列表data[‘employees’]的第一个元素中，请在空格处补全代码完成该元素的访问。

**data = {**

**'employees': [**

**{'name': 'Alice', 'age': 30},**

**{'name': 'Bob', 'age': 25}**

**]**

**}**

**name\_of\_first\_employee = data['employees'][\_\_\_0\_\_]['name']**

**7.**分析以下代码，当调用MyClass.static\_method()时，输出是（**This is a static method**）。

**class MyClass:**

**@staticmethod**

**def static\_method():**

**print("****This is a static method")**

**MyClass.static\_method()**

**8.**在Python的类定义中，如果希望某个方法仅作为类方法而非实例方法，并且能够访问类级别的属性，则需要在该方法前加上装饰器\_

\_\_\_**@classmethod**\_\_。

**9.**在使用Python进行网页爬虫时，\_\_requests\_\_\_\_库是一个非常流行的选择，它提供了简单易用的API来发送HTTP请求并处理响应。

**10.**在Python中，\_\_ **Matplotlib**\_\_\_\_库是一个用于创建静态、交互式和动态图形的可视化库，它支持多种图表类型，是一个通用基础的可视化库。

11.假如你设计了一个Python文件A.py, 其中要用到求一个角的三角函数值，那在A.py中引入哪个模块，具体的语句是（import math）。

12. input()函数用于接收用户键盘输入的数据，假如在提示信息：请输入小明的身高：后输入身高（以米为单位），并存在变量h中，具体的语句是：

（h = float(input("请输入小明的身高："))）。

13.执行下面代码的结果是（ 4 ）

word = 't'

string = 'hljit.edu.cn'

result = string.find(word)

print(result)

14. 李华编写一段程序代码如下，在运行时出错，错误提示为：NameError: name 'num\_two' is not defined

1：num\_one = 1

2：result = num\_one + (num\_two)

3：print(result)

如果李华不想多写一行num\_two=2的语句，他应该怎么更改代码才能保证程序运行打印出来的结果是3，请把更改的代码行号和更改的内容写在括号里。

（result = num\_one + (num\_two: =2)）

15.李华想设计一个学生成绩管理系统，需要存储一个学生信息如下：学号：001，姓名：李华，课程名：python程序设计，成绩：95。他应该选择（ 字典 ）数据类型进行学生信息存储。

16. 为了过滤用户从键盘输入的评论中的敏感词语，李华书写了下面的一段代码，把输入中带有的敏感词都用'\*'表示，那么括号中应填入什么内容。

sensitive\_list = ['敏感词1','敏感词2','敏感词3']

test\_sentence = input('请输入一段话:')

for line in sensitive\_list:

if line in test\_sentence:

test\_sentence =（ test\_sentence.replace(line, “\*”)）

print(test\_sentence)

17.李华在程序中定义了一个sum函数对用户输入的数据进行求和，但是用户输入的数据个数不定，所以他在定义sum函数时，sum函数的参数应该是（\*args）。

18. 李华定义3个类，一个是House类，一个是Car类具体代码如下，House类提供一个居住方法，Car类提供一个行驶方法。还要定义一个房车类，既有居住方法又有驾驶方法。请将横线处的代码补充完整。

class House(object):

def live(self):

print("供人居住")

class Car(object):

def drive(self):

print("行驶")

class TouringCar( House,Car ):

pass

19.网络爬虫程序涉及HTTP请求，Python计算生态通过（ requests ）库，提供了简单易用的类HTTP协议。

20.turtle是Python（ 内置 ）的一个标准模块，它提供了绘制线、圆以及其它形状的函数，可以通过它绘制简单的数据可视化图形。

**二、程序填空题**

**1.**请补全以下Python代码，使其实现一个“银行账户”类（BankAccount）。该类应具有账户余额（balance）和账户名（account\_name）两个属性，并包含以下三个方法：

deposit(amount)：向账户存款，amount为存款金额。

withdraw(amount)：从账户取款，amount为取款金额。如果余额不足，应打印“余额不足，取款失败”。display\_balance()：显示当前账户余额。

**class BankAccount:**

**def \_\_init\_\_(self, account\_name, initial\_balance):**

1. **# 补全构造方法**

**self.account\_name = account\_name # 初始化账户名**

**self.balance = initial\_balance**

**（2）# 填写deposit方法**

**def deposit(self, amount):**

**self.balance += amount**

**（3） #填写withdraw方法**

**def withdraw(self, amount):**

**if self.balance >= amount:**

**self.balance -= amount**

**else:**

**print("余额不足，取款失败。")**

**（4）# 填写display\_balance方法**

**def display\_balance(self):**

**print(f"当前余额为：{self.balance}元")**

**2.**补全下面程序，使其实现：输入三角形三条边判断能否构成三角形，若能构成三角形则输出：“可以构成三角形，其周长为：\*”，其中“\*”为输入的三边后求的周长值。否则引发异常，输出：“不能构成三角形。”

class MessageError(Exception):

def \_\_init\_\_(self):

(1) self.no\_message = "不能构成三角形。"

class Triangle:

def composed\_triangle(self):

side\_length\_one = int(input("请输入第一个边长：\n"))

side\_length\_two = int(input("请输入第二个边长：\n"))

side\_length\_three = int(input("请输入第三个边长：\n"))

try:

if not (（2）1+2>3 and 1+3>2 and 2+3>1 注：（1，2，3分别表示上述三个边长，懒得写了） ):

（3）raise MessageError() # 引发异常

except MessageError as e:

print(e.no\_message)

else:

（4）p=1+2+3

print(f"可以构成三角形，其周长为：{p}")

triangle = Triangle()

triangle.composed\_triangle()

3.需求：现有四大名著之一《西游记》的文本文件，名字为xiyouji.txt。要完成对该文本的分词统计，并且把高频词的统计结果以词云的方式展示出来。其中部分代码已经给出，需要你按照要求补齐划线处的代码。

请根据需求和提示填空。

import jieba #导入中文分词库和词云库

（1） from wordcloud import WordCloud导入词云库

import re

import numpy as np

import matplotlib.pyplot as plt

txt= （2）open('xiyouji.txt').read()# 打开并读取“xiyouji.txt”

txt2=re.sub(r"[^\u4e00-\u9fa5]","",txt) # 替换文本中的非文字符号为空格

txt3= （3）jieba.lcut(txt2) #txt3是对txt2进行分词后返回的一个可迭代对象

txt4={} #存放每个词出现次数的字典

for i in txt3:

if i not in txt4:

txt4[i]=1

else:

txt4[i]+=1

txt5=sorted(txt4.items(),key=lambda x:x[1],reverse=True) #txt5存放排序后的各词按词频降序排列

txt6={}

for word,count in txt5:

txt6[word]=count

list1=["了","和","等","在","到","得","是","的","与","这","又","就"] #设置排除字词列表

#删除排除词列表中的词语

（ 4）for i in list1:

del txt6[i]

# 词云绘图要求：背景为白色，字体为：simhei.ttf，最大词数为50，最大字号为100，宽度和高度都为500

wordcloud=WordCloud(（5 ）font\_path='simhei.ttf', # 字体路径

background\_color='white', # 背景颜色

max\_words=50, # 最大词数

max\_font\_size=100, # 最大字号

width=500, # 宽度

height=500 # 高度

).generate\_from\_frequencies(txt6)

plt.figure(figsize=(8,8)) #设置画布的大小为8英寸\*8英寸

plt.imshow(wordcloud,interpolation='bilinear')

plt.axis('off')

plt.show()

txt.close() #关闭文件

4.需求：用户输入用户名，对用户名称的要求有两点。一是长度为8-12。二是只能包含数字或字母不能有其他字符，在满足要求一的基础上满足要求二。如果不满足要求，则抛出相应的自定义异常，并捕获该异常，如果输入符合要求，则打印输出“输入成功”。

提示：自定义判断用户名长度异常类名称：ShortInputError，有两个属性密码长度属性：length，密码最小长度属性atleast。自定义判断用户名合法性的异常类名称：RulerError请根据需求和提示填空。

class ShortInputError(Exception):  
 def \_\_init\_\_(self, length, min,max):  
 self.length= length

1. self.min= min# 用户名最小长度

self.max= max   
class RulerError(Exception):  
 def \_\_init\_\_(self,name):  
 self.name= name   
try:  
 text = input("请输入用户名：")  
 if ② len(text) < 8 or len(text) > 12: #判断长度是否合法  
 raise ShortInputError(len(text), 8，12)  
except ShortInputError as result:  
 print("ShortInputError：输入的长度是%d，长度应是%d至%d之间" %

(result.length, result.min, result.max)) #补全该语句

else:  
 try:  
 if not (text.isalnum()):  
 ④raise RulerError(text)# 抛出异常  
 except RulerError:  
 print("用户名中有除了数字和字母以外的非法字符")  
 else:  
 ⑤ print("输入成功")#符合要求后的输出

三、程序设计题

1.新生入学后，每个班级由n位同学，每位同学来自的省份不全相同，现在想统计一个班级中同学的省份来源数据。请选择一种合适的数据类型接收从键盘输入的每位同学的省份数据，并利用这种数据类型便利操作编码实现删除重复省份的信息，得到该班级学生不重复的省份数据，并打印显示，请编写代码实现上述功能。（提示：注意判断输入学生人数和输入省份数据个数不对应时应给出提示“你输入的省份数量和学生数据不一致，程序结束”，结束程序。）

例如1：

输入：

请输入班级同学人数：10

请依次输入每个同学的省份：黑龙江,辽宁,福建,河北,辽宁,河南,河北,福建,辽宁,河北

输出：

该班级学生的省份如下：{'黑龙江', '辽宁', '福建', '河北', '河南'}

例如2：

请输入班级同学人数：3

请依次输入每个同学的省份：黑龙江,辽宁

你输入的省份数量不是3个，程序结束

num = int(input("请输入班级同学人数:"))

sfs = input("请依次输入每个同学的省份:")

sfl = sfs.split(",")

if len(sfl) != num:

print(f"你输入的省份数量不是{num}个，程序结束")

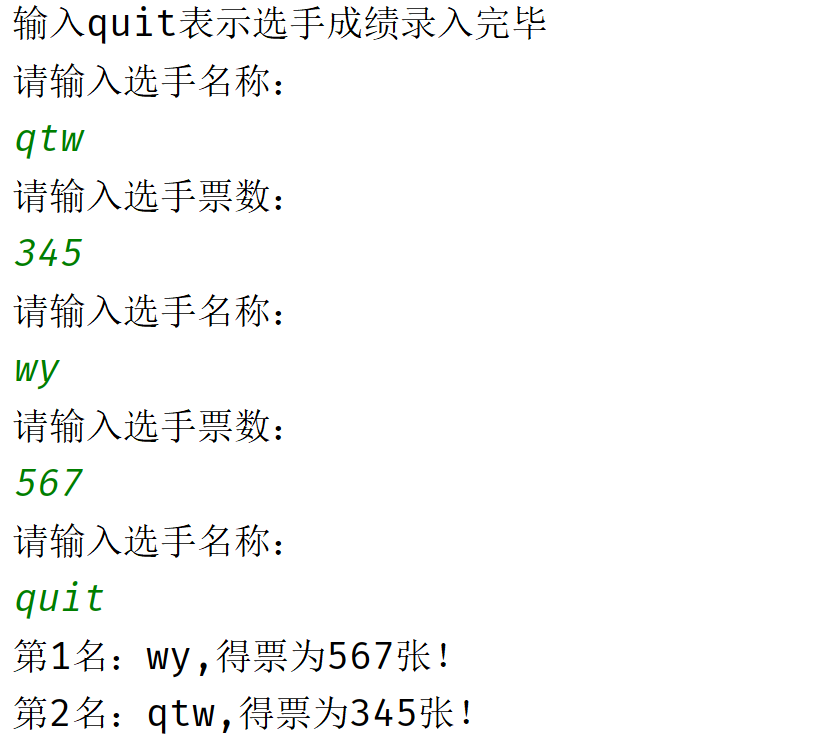
else:

new\_s = set(sfl)

print(f"该班级学生的省份如下: {new\_s}")

**2.**学校举行主题为“强国有我”的迎国庆演讲比赛。比赛采用计票机制，选手获得的票数越多，排名就越靠前。要求编写程序，接收选手的姓名和票数，按照排名降序输出名次，选手姓名，得票数。

提示：系统运行后显示：“请输入选手姓名：”，用户输入姓名后，系统提示“请输入选手获得票数：”输入票数后，再循环出现提示:“请输入选手姓名：”,用户输入姓名后，系统再提示“请输入选手获得票数：”当录入完毕所有选手信息后，再提示：“请输入选手姓名：”时录入“quit”则程序退出。最后系统运行结果如下图所示：



player\_info = {} # 初始化字典存储选手信息

li = []

print("输入quit表示选手列表录入完毕")

while True:

name = input("请输入选手名称: \n")

if name == 'quit':

break

score = int(input("请输入选手分数: \n"))

player\_info[name] = score

items = player\_info.items()

for j in items:

li.append([j[1],j[0]])

li.sort() # 根据分数降序排序

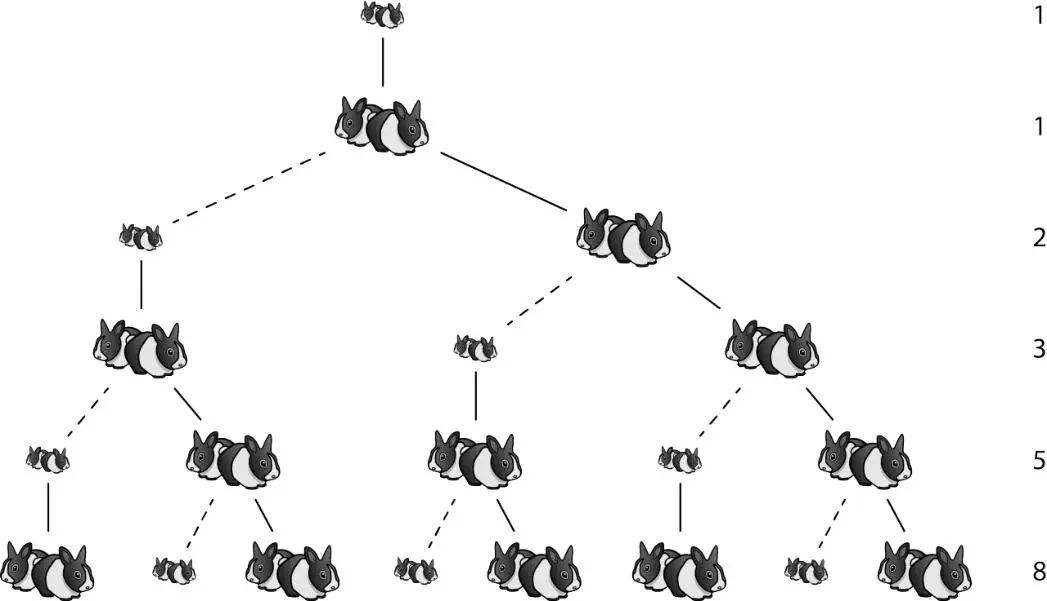
count = len(li) - 1

for i in range(1,len(li) + 1): # 循环次数应该是列表长度

print(f"第{i+1}名, {li[count][1]}, 得票为{li[i][0]}张！")

count -= 1

**3.**兔子一般在出生两个月之后就有了繁殖能力，每对兔子每月可以繁殖一对小兔子，假如所有的兔子都不会死。请你根据下图提示，设计一个递归函数，实现根据输入的月份计算到该月时兔子总数量，并打印输出。请编写代码实现上述功能。



例如：

输入：请输入月数:4

输出：到第4月时，兔子有6只。

def fibonacci(month):

if month == 1 or month == 2:

return 1

else:

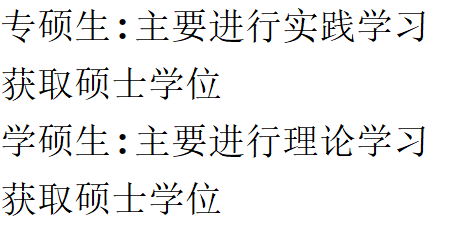
return fibonacci(month-1) + fibonacci(month-2)

num = int(input("请输入月数:"))

result = fibonacci(num) \* 2

print(f"到第{num}月时，兔子有{result}只")

**4.**在硕士研究生阶段的学生分专硕和学硕。‌专硕更注重实践，‌学硕更注重理论。但他们都以获得硕士学位为目的。请你编写Student类，具有类型属性：type（学硕或专硕），具有行为：study(学习 )和get\_master(获得硕士学位)。定义Pm（专硕)类和Am(学硕)类继承Student类。Pm（专硕)类和Am(学硕)类学习的行为不同，Pm（专硕)类是主要进行实践学习，Am(学硕)类是主要进行理论学习；但获得学位是一致的，均为获取硕士学位。请编写代码实现上述类并实例化一个专硕生和一个学硕生，输出以下结果：



class Student:

leixing = “”

def \_\_init\_\_(self, lx):

self.leixing = lx

def xuexi(self):

pass

def get\_xuewei(self):

return "获取硕士学位"

class Zhuanshuo(Student):

def xuexi(self):

return "通过实践学习"

class Xueshuo(Student):

def xuexi(self):

return "通过理论学习"

# 创建专业硕士和学术硕士的对象

z1 = Zhuanshuo("专硕生")

x1 = Xueshuo("学硕生")

# 打印专业硕士的信息

print(z1.leixing + ":" + z1.xuexi() + "\n" + z1.get\_xuewei())

# 打印学术硕士的信息

print(x1.leixing + ":" + x1.xuexi() + "\n" + x1.get\_xuewei())

5.在中华传统文化中，百家争鸣的各派哲学思想一直影响着中华民族的发展，其中最具典型的为儒家和道家。儒家的宗师为孔子，代表作为《论语》 ，道家的宗师为老子，代表作为《道德经》。儒家和道家均关注个人的修养和道德，但他们又有不同的修身方式。

（1）编写Zhexue(哲学)类，具有属性：派别（paibie），宗师（zongshi）,代表作（dabiaozuo）；具有行为关注（guanzhu）和行为修身（xiushen）。

（2）定义Rujia（儒家)类和Daojia(道家)类继承Zhexue(哲学)类。

（3）Rujia（儒家)类和Daojia(道家)类，的修身行为不同，Rujia（儒家)是“刻意修为，成为圣人”，Daojia(道家)是“道法自然，天人合一”；但是他们对人生有共同的关注，均关注个人的修养和道德。

在考虑类的继承的基础上，合理设置各类的属性和方法，并实例化儒家类和道家类，进行代码编写，输出以下结果：

儒家->宗师：孔子,代表作：《论语》

关注个人修养和道德

修身方式：刻意修为，成为圣人

道家->宗师：老子，代表作《道德经》。

关注个人修养和道德

修身方式：道法自然，天人合一。

class Zhexue:

def \_\_init\_\_(self, pb, zsh, dbz):

self.paibie = pb # 派别

self.zongshi = zsh # 宗师

self.daibiaozuo = dbz # 代表作

def getinfo(self):

print(f"{self.paibie}宗师:{self.zongshi}, 代表作:{self.daibiaozuo}")

def guanzhu(self):

print("关注个人修养和道德")

class Rujia(Zhexue):

def xiushen(self):

print("修身方式：刻意修为，成为圣人")

class Daojia(Zhexue):

def xiushen(self):

print("修身方式：道法自然，天人合一")

# 创建儒家和道家的实例

r = Rujia("儒家", "孔子", "《论语》")

r.getinfo()

r.guanzhu()

r.xiushen()

d = Daojia("道家", "老子", "《道德经》")

d.getinfo()

d.guanzhu()

d.xiushen()

6.假设某网站只允许用户上传jpg、png和bmp格式的文件，要求编写代码，自定义异常类，通过异常捕获语句实现用户上传头像文件格式检测的功能。要求输入文件全名。文件名称.扩展名，具体输入格式如下例如1，例如2所示。请按要求编写程序代码实现上述功能。

例如1：

输入：

请输入上传图片的名称（包含格式）：pic.jpg

输出：

上传成功

例如2：

输入：

请输入上传图片的名称（包含格式）：pic.gif

输出：

仅支持jpg/png/bmp格式

class FileTypeError(Exception):

def \_\_init\_\_(self, err="仅支持 jpg/png/bmp 格式"):

super().\_\_init\_\_(err)

file\_name = input("请输入上传图片的名称（包含格式）: ")

try:

if file\_name.split(".")[-1] in ["jpg", "png", "bmp"]:

print("上传成功")

else:

raise FileTypeError

except Exception as error:

print(error)