考试时间：90 分钟

考试题型：

1. 选择题（每小题 2 分，共 20 分）
2. 判断题（每小题 1 分，共 10 分）
3. 填空题（每空 1 分，共 10 分）
4. 编程题（每小题 10 分，共 30 分）
5. 应用题（每小题 15 分，共 30 分）

重点内容：

一、选择题

（一）第 2 章 语法基础

1. P14：在 Python 程序设计语言中，用于输入和输出的函数分别是 input()和 print()。
2. P17：标识符命名规则，例如 student_sex，get_msg 都是合法标识符。
3. P19：Python 语言中基本的数据类型主要有整型（int）、浮点型（float）、布尔型（bool）、字符串类型（str）。
4. P26：算数运算符，例如“//”表示整除，“%”表示取模，“/”表示除以。

（二）第 3 章 流程控制

1. P45：Python 中，range()函数用于生成整数数字序列，例如想要使用 range()获取数字 1~5，正确的写法是 range(1,6)。
2. P44-50：Python 循环结构，
   ① 遍历循环中的数据结构可以是字符串、列表、元组和 range()函数等。
   ② 可以通过 for、while 等关键字来构建循环结构。
   ③ 关键字 continue 用于结束本次循环 ，break 同于跳出当前循环层。

（三）第 4 章 常用数据结构

1. P59：列表元素的访问，例如 a=[1,2,3,4,5,6,7,8,9,10]，则输出列表 a 的最后一个元素的语句是 print(a[9])。
2. P63：列表的常见方法，
   ① insert()：在列表的指定位置插入一个元素，x=[1,2,3]，x.insert(1，4)得到[1, 4, 2, 3]。
   ② pop()：删除列表中指定位置的元素。例如 a[1, 2, 3, 4, 5]执行 a.pop(3)得到[1, 2, 3, 5]。
   ③ remove()：删除列表中第一次出现的指定元素。例如 a[1, 2, 3, 4, 5]执行 a.remove(4)得到[1, 2, 3, 5]。
3. P87-89：字典，
   ① 字典是一种映射类型，由若干”键：值”组成，其中键必须为不可变类型，例如：元组、字符串、整数都可以做为键。
   ② 字典的创建，例如 a_dict = {}, a_dict 是空字典。
   ③ 字典的访问，例如通过”字典对象[键]”获取对应值时，如果键不存在，则报错。
   ④ 字典的常见方法，例如字典的 get()方法，如果不存在该键，则返回默认值。

（四）第 5 章 函数

1. P99-106：参数类型与参数传递

例：def test(b=2, a=4):

```
        global z
        z += 3*a + 5*b
        return z
    z = 10
    print(z, test(4, 2))
```

输出结果：10 36

2. P108：变量作用域，例如使用 global 关键字可以将函数内部变量声明为全局变量。

（五）第 7 章 常见库操作

1. P134:数学库 math，sqrt(浮点数)获取浮点数的平方根，pow(底数,指数)求幂。
2. P136：随机库 random，random 库的方法很多，其大部分方法都具有随机性特征。

（六）第 8 章 文件操作

1. P148-149：文件读写，

① 有一个包含中文字符的 test.txt 文件，现在希望以只读的方式打开它，实现的操作是 open("test.txt", mode="r", encoding="utf-8")。

② open() 方法用于打开一个文件，并返回文件对象。open() 方法的声明如下：open(file, mode='r', buffering=None, encoding=None, errors=None, newline=None, closefd=True)，file: 表示文件名或文件路径的字符串；mode: 文件打开的模式，如读、写、追加等，默认为读。

③ 二进制文件和文本文件的操作步骤都是“打开-操作-关闭”。

④ open()打开文件之后，文件的内容并没有在内存中。

⑤ 文件读写之后，要调用 close()才能确保文件被保存在磁盘。

（七）第 9 章 面向对象编程

1. P171：以 2 个下画线开始的类属性属于私有属性。

（八）第 10 章 数据库操作

1. P197：Python 操作数据库 SQLite，Python 操作数据库的基本操作流程，

① 导入模块：sqlite3；

② 连接数据库得到 Connection 对象，sqlite3.connect(文件名)；

③ 获取 Cursor 对象，Connection 对象.cursor；

④ 执行数据库的增删改查操作，Cursor 对象.execute(sql 语句)；

⑤ 提交数据库操作，Connection 对象.commit() ；

⑥ 关闭 Cursor，Cursor 对象..close()；

⑦ 关闭 Connection，Connection 对象.close()；

（九）第 11 章 NumPy 入门与实践

1. P214、235：NumPy 常用方法，例如：

```
import numpy as np
a=np.arange(12).reshape((3,4))
print(a.mean())
```

的输出结果为： 5.5

（十）第 12 章 Pandas 入门与实践

1. Pandas 入门，

① Pandas 中 loc 是显式索引访问，iloc 是隐示索引访问。

② Pandas 中的 Series 和字典数据结构最相似。

③ Pandas 中的 Index 对象可以包含重复值。

④ Pandas 中的 merge()方法，默认是内连接，使用共同的列进行关联。

（十一）第 13 章 数据可视化之 matplotlib

1. P292：matplotlib.pyplot 的常用方法中，subplot(a, b, c)方法 c 位置的参数所代表的含义是这是第几张图。

二、判断题

（一）第 2 章 语法基础

1. 语法、变量和注释。

① Python 语言采用空格或制表符来表明每行代码的层次关系。

② 当出现语法错误时，必须在程序执行前改正错误，否则程序无法运行。

③ Python 中，常说的变量类型是指变量指向的对象的数据类型。

④ 标识符的命名不能以数字开头。

⑤ Python 代码的注释，可以使用”#”，三个单引号（’’’）），或者三个双引号（”””）。

2. 运算符

① 身份运算符 is 用于判断两个对象是否为同一对象，比较两个对象的内存位置是否一致。

（二）第 4 章 常用数据结构

1. 字符串是有序的、不可变的序列，和列表、元组一样都支持索引、切片操作。
2. Python 字典不支持索引。
3. Set 集合中的元素不能重复。

（三）第 5 章 函数

1. 所有通过 lambda 表达式实现的功能都可以通过相应的函数实现，反之则不一定。
2. 定义函数时，可以没有参数，但必须有一对小括号。
3. 在函数内部对变量赋值时，如果没有对变量进行任何声明，这个变量一定是局部变量。

（四）第 6 章 异常处理

1. P123：异常处理结构

① 在异常处理结构中，无论是否发生异常都会执行 finally 子句。

② try 子句后面可以有多个 except 子句，分别用来处理不同类型的异常，但最多只有一个 except 子句会被执行。

（五）第 12 章 Pandas 入门与实践

1. DataFrame 常用方法中，isna()、isnull()可以用来判断元素是否为缺失值。
2. P255-256：显示索引和隐式索引。
3. P267：DataFrame 中 join()方法用于将其他 DataFrame 中的列合并到当前 DataFrame 中，默认采用左连接。

（六）第 14 章 人工智能之 scikit-learn 入门与实践

1. P316：机器学习常见算法有回归算法、决策树算法、贝叶斯算法、随机森林、神经网络、支持向量机、聚类算法等。

三、填空题

（一）第 2 章 语法基础

1. bool(x)函数，可以将 x 转换成 bool 型，那么，bool(10)的返回值是 True。
2. Python 的数据类型分为整型、浮点型、字符串、布尔型等类型。
3. 假设 x=2,表达式 x**3 的值是 8 ,表达式 13//2/x 的值是 3.0 。

4. 设 s=‘abcdefg’，则 s[3]值是 ‘d’ ，s[3:5]值是‘de’。

（二）第 3 章 流程控制

1. 循环控制语句主要包括 break 语句和 continue 语句，break 语句 用于终止或中断当前循环层语句， continue 语句 用于终止当次循环。

2. range(start,stop[,step])函数,用于生成整数数字序列，start 的默认值是 0 ，step 的默认值是 1 ,则 list(range(1,7,2))转换成的列表是 [1,3,5] 。

（三）第 4 章 常用数据结构

1. 集合是无序 序列。

2. 假设 s=’ abcdefg’，则 s[3]的值是 ’ d’ ，s[2:5]的值是 ’ cde’ ，s[:-3:-1]的值是 ’ gf’ 。

3. Python 序列类型包括 元组、列表 、字符串 三种； 字典 是 Python 中唯一的映射类型。

（四）第 5 章 函数

1. 常见的函数参数的传递类型： 关键字参数 、位置参数、 默认参数 和可变参数等。

2. P113：map()函数是 Python 的内置函数，用于多次调用某一函数，并将可迭代对象中的元素作为实参传入，最终返回结果为函数运行结果的迭代器。

（五）第 7 章 常见库操作

1. Python 中导入模块要使用关键字 import 。

2. Python 的数学库 math 中，ceil()和 floor()分别用来对浮点数进行向上取整和向下取整。

（六）第 8 章 文件操作

1. 文件路径分两类，其中 绝对路径 是你的主页上的文件或目录在硬盘上真正的路径，是从盘符开始的路径； 相对路径 是相对于当前文件的路径。

2. with 是一种上下文资源管理器，对资源进行自动管理。

（七）第 9 章 面向对象编程

1. 类中定义的方法大致分为三类：实例方法、 类方法 和 静态方法 。

2. P176：如果在类定义时没有指定父类，则默认其父类为 object，object 是所有类的基类。

四、编程题

（一）第 4 章：列表、字符串。

输入一个字符串，输出该字符串中出现频数最高的字符，如果有多个字母出现的频数并列最高，则输出多个字母。

```
a_str = input("请输入一个字符串：")
a_list = []
b_list = []
c_max = 0
for c in a_str:
    if c not in a_list:
        a_list.append(c)
        c_current = a_str.count(c)
        if c_current > c_max:
            b_list.clear()
            b_list.append(c)
            c_max = c_current
        elif c_current == c_max:
            b_list.append(c)
        else:
            continue
print(b_list)
```

（二）P94 课后练习 4.6：存在字符串“I，love，python”，统计该字符串中各种字符出现的次数，并将统计结果从高到低进行排序，最终打印排序后的信息。每行效果如下：

**** 字符出现次数为：****次

```
a_str = "I，love，python"
a_dict = {}
for i in a_str:
    old_num = a_dict.get(i, 0)
    a_dict[i] = old_num + 1
result = sorted(a_dict.items(), key=lambda item: item[1], reverse=True)
for res in result:
    print(res[0], "字符出现的次数为：", res[1])
```

（三）P145 课后练习 7.7：编写一个程序，随机生成 1000 个字母，包含大写字母和小写字母，然后统计各个字母出现的次数，统计时忽略字母的大小写，最后将统计结果按照字母出现的次数从高到低排序输出。

```
import random as ra
from collections import Counter
low = [chr(x) for x in range(97, 97 + 26)]          #根据 ascII 码把数字转化为字母
up = [chr(x) for x in range(65, 65 + 26)]
low.extend(up)              #将大写字母与小写字母放入一个列表里面
result = ra.choices(low, k=1000)       #随机抽取 1000 个
s = "".join(result)           #把字符序列转换成一个字符串
s = s.lower()                #将所有随机生成的字母变为小写
c = Counter(s)                  #以集合的形式统计结果
print(c.most_common())
for key, value in c.most_common():                    #c.most_common 生成列表，是有序的。
    print(key, ":", value)
```

（四） P127 课后练习 6.1：编写一个程序，提示用户输入一个整数，如果输入的不是整数，则让用户重新输入，直到是一个整数为止。

例如：第一次输入 abc，第二次输入 12.5，第三次输入 6，执行效果如下:

请输入一个整数：abc

输入不符合要求，请重新输入！

请输入一个整数：12.5

输入不符合要求，请重新输入！

请输入一个整数：6

输入正确，你输入的整数为：6

```
def test():
    while True:
        try:
            num = int(input("请输入一个整数："))
            print("输入正确，你输入的整数为：", num)
            return
        except ValueError:
            print("输入不符合要求，请重新输入！ ")
test()
```

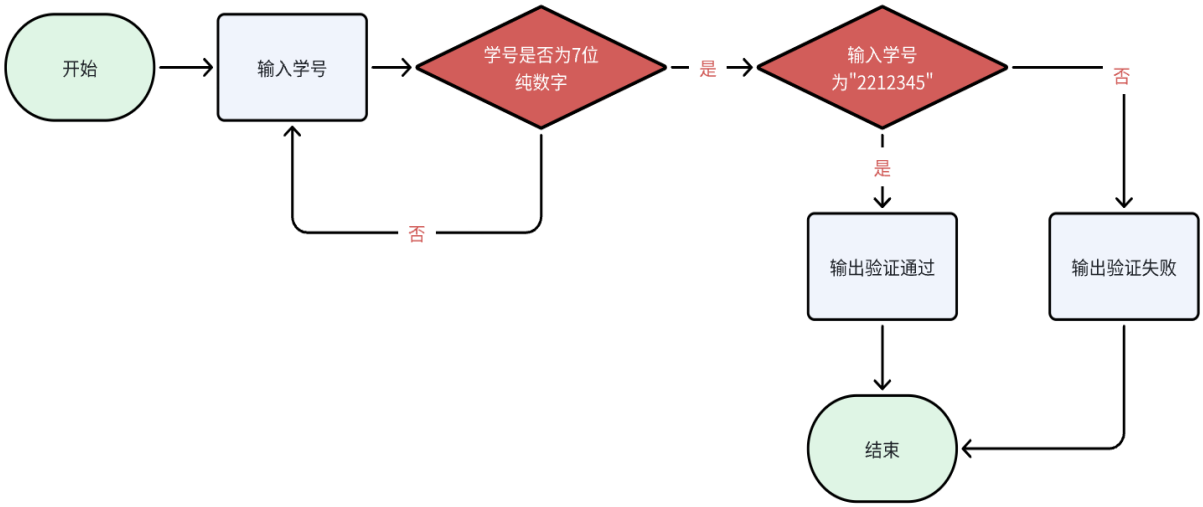
（五）编写函数实现如下功能，对传递的一组数据进行操作，调整数据的位置，使得所有的奇数位于前半部分，所有的偶数位于后半部分，并保证奇数和奇数，偶数和偶数之间的相对位置不变，输出调整后的数据。

例如：原始数据为：[9，6，7，3，1，8，4，3，6]，

则调整后的数据为：[9，7，3，1，3，6，8，4，6]。

```
def adjust_position(x):
    even_num = []
    odd_num = []
    for n in x:
        if n % 2 == 0:
            even_num.append(n)
        else:
            odd_num.append(n)
    odd_num.extend(even_num)
    print(odd_num)
adjust_position([9, 6, 7, 3, 1, 8, 4, 3, 6])
```

（六）P127 课后练习：编写程序实现账号验证功能（要求：定义函数，调用函数）



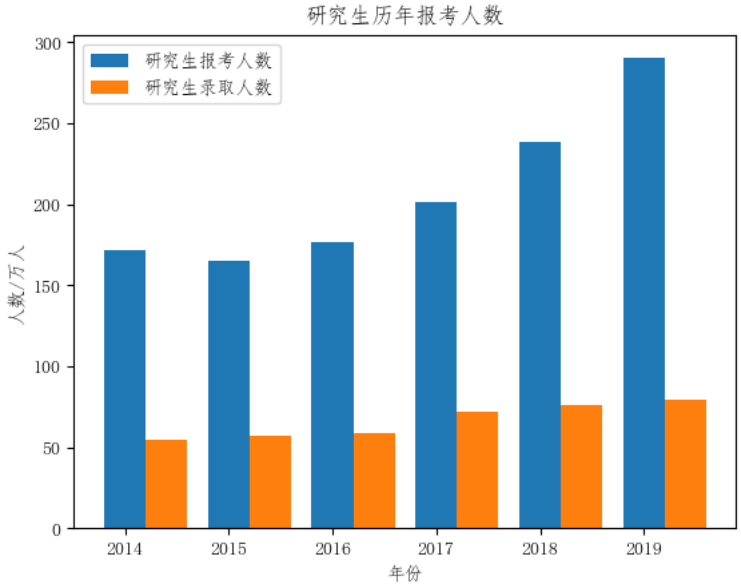

```
def verify_no():
    while True:
        try:
            input_no = input("请输入学号：")
            if len(input_no) == 7 and int(input_no):
                if input_no == '2311101':
                    print("验证通过")
                else:
                    print("验证失败")
                break
            else:
                print("输入不符合要求，请重新输入！")
                continue
        except ValueError:
            print("输入不符合要求，请重新输入！")
verify_no()
```

五、应用题

（一）P287 课后练习 12.10

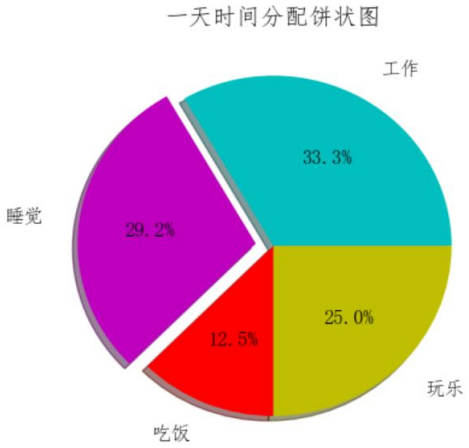
```
import pandas as pd
import numpy as np
#第一题
data_1 = pd.read_excel("exer_1.xlsx")
data_2 = pd.read_excel("exer_2.xlsx")
data_3 = data_1.join(data_2.set_index("姓名"), on="姓名")
print(data_3)

#第二题，按总分排序
print(data_3.sort_values(["总分"],ascending=False))

#第三题
print(data_3[(data_3["语文"] < 60)|(data_3["数学"] < 60)|(data_3["英语"] < 60)])

#第四题
print("语文科目最高分是：",data_3["语文"].max())
print("语文科目最低分是：",data_3["语文"].min())
print("语文科目平均分是：",round(data_3["语文"].mean(),2))

#第五题
print("3 班女生语文的平均分是：")
print(round(data_3[(data_3["班级"] == "3 班") & (data_3["性别"] == "女")]["语文
"].mean(),2))

#第六题
print(round(data_3.groupby("班级").agg([np.max, np.min, np.mean])[["数学"]]))

#第七题
print(round(data_3.groupby("性别").agg([np.max, np.min, np.mean]).loc["男":"男"]))
```

（二）P303 绘制条形图



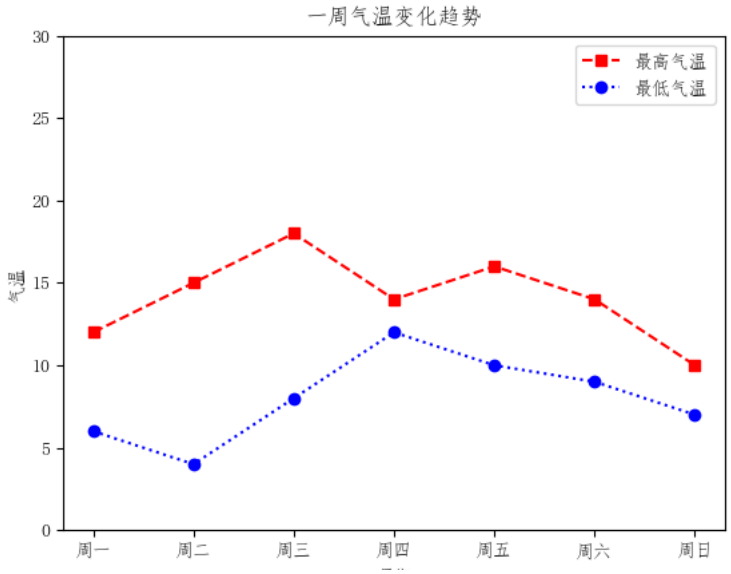

```
import matplotlib.pyplot as plt
import numpy as np
plt.rcParams["font.family"] = "FangSong"
plt.rcParams["font.size"] = 10
nums = [172, 164.9, 177, 201, 238, 290]
luqu_nums = [54.87, 57.06, 58.98, 72.22, 76.25, 79.3]
x = range(0, len(nums))
x_ticks = ["2014", "2015", "2016", "2017", "2018", "2019"]
plt.xticks(x, x_ticks)
plt.bar(x, nums, width=0.4, label = "研究生报考人数")
plt.bar([i+0.4 for i in x], luqu_nums,width=0.4, label = "研究生录取人数")
plt.title("研究生历年报考人数")
plt.xlabel("年份")
plt.ylabel("人数/万人")
plt.legend()
```

（三）P304 绘制饼图



```
import matplotlib.pyplot as plt
plt.rcParams["font.family"] = "FangSong"
plt.rcParams["font.size"] = 12
labels = ["工作", "睡觉", "吃饭", "玩乐"]  #活动标签
hours = [8, 7, 3, 6] #时间分配
colors = ["c", "m", "r", "y"] #各部分颜色
plt.pie(hours, labels=labels, colors=colors, shadow=True, explode=(0,0.1,0,0), autopct="%.1f%%", labeldistance=1.2)
plt.title("一天时间分配饼状图")
plt.show()
```

（四）P297 绘制线型图




```
import matplotlib.pyplot  as plt
import numpy as np
plt.rcParams["font.family"] = "FangSong" #便于中文显示

x = ["周一", "周二", "周三", "周四", "周五", "周六", "周日"]
highest = [12, 15, 18, 14, 16, 14, 10] #最高气温
lowest = [6, 4, 8, 12, 10, 9, 7] #最低气温
plt.ylim(0, 30)  #设置 y 轴的取值范围
plt.plot(x, highest, "rs--", label="最高气温")
plt.plot(x, lowest, "bo:", label="最低气温")
plt.title("一周气温变化趋势")  #标题
plt.xlabel("星期")
plt.ylabel("气温")
plt.legend()
plt.show()
```