**1**

**一、Python基础练习（至少选择5题）**

**1. 编程**，创建一个包含200-300之间所有素数列表，显示该列表；随后只保留该列表前5个数字，删除其余内容并打印输出相应结果；再将每个元素值加上100，显示列表内容。

def is\_prime(n):  
 if n <= 1:  
 return False  
   
 for i in range(2, int(n\*\*0.5) + 1):  
 if n % i == 0:  
 return False

return True  
  
primes = []  
for i in range(200, 300):  
 if is\_prime(i):  
 primes.append(i)

# 显示包含200-300之间所有素数列表  
print(primes)

# 只保留列表前5个数字，删除其余内容  
primes = primes[:5]  
print(primes)

# 再将每个元素值加上100  
primes = [x + 100 for x in primes]  
print(primes)

**结果：**

文本

描述已自动生成

**2．编程**，根据需要定义列表，并完成数的输入，最后打印输出最终列表

1）创建一个长10000的列表blist（列表元素内容为1开始的整数）

2）创建一个长10000的列表clist（列表元素内容均为1）

3）将blist和clist列表连接为1个新列表后，删除所有数字1的元素。

import numpy as np  
  
# 创建一个长10000的列表blist（列表元素内容为1开始的整数）  
blist = np.arange(1, 10001)  
  
# 创建一个长10000的列表clist（列表元素内容均为1）  
clist = np.ones(10000, dtype=int)  
  
# 将blist和clist列表连接为1个新列表删除所有数字1的元素  
new\_list = np.concatenate((blist, clist))  
new\_list = new\_list[new\_list != 1] #删除所有数字1的元素  
print(new\_list)

后，删除所有数字1的元素。  
new\_list = np.concatenate((blist, clist))  
new\_list = new\_list[new\_list != 1]  
print(new\_list)

**结果：**

文本

描述已自动生成

**3.编程**：开发敏感词语过滤程序，提示用户输入评论内容，如果用户输入的内容中包含特殊的字符：敏感词列表list1 = [“贾玲”, “热辣滚烫”,“大碗娱乐”, “减肥100斤”] 则将用户输入的内容中的敏感词汇替换成等长度的（贾玲就替换\*\*），并添加到一个列表中；如果用户输入的内容没有敏感词汇，则直接添加到上述的列表中。

import numpy as np  
  
list1 = ['贾玲', '热辣滚烫','大碗娱乐','减肥100斤']  
comment = input("请输入评论内容：")  
for keyword in list1:  
 comment= comment.replace(keyword,'\*'\*len(keyword))  
print(comment)

**结果：**

文本

描述已自动生成

**5.编程**，构建字典：假如有如下数值的列表[68,55,49,88,75,66,37,88,29,90,61,42,83,73],将其中所有大于等于60的数保存至字典的第一个键’jg’中，将小于60的数保存至第二个键’bjg’中即：{'jg':大于等于60的数,'k2':小于60的数}。

import numpy as np  
list = [68,55,49,88,75,66,37,88,29,90,61,42,83,73]  
arr = np.array(list)  
  
# 创建一个空字典  
dictionary = {}  
  
# 使用布尔索引找出数组中大于等于 60 的数，并将它们保存到字典的 'jg' 键中  
dictionary['jg'] = arr[arr >= 60]  
  
# 使用布尔索引找出数组中小于 60 的数，并将它们保存到字典的 'bjg' 键中  
dictionary['bjg'] = arr[arr < 60]  
  
print(dictionary)

**结果：**

文本

描述已自动生成

**7.集合操作**

1）创建空集合a\_Set

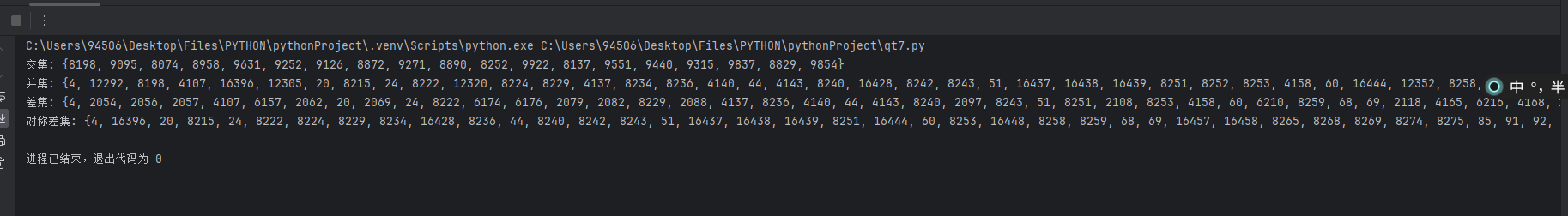
2）创建一个集合b\_Set含有1000个元素，每个元素是0-10000之间的一个随机数。

3）创建一个集合c\_Set含有1000个元素，每个元素是8000-20000之间的一个随机数。

4）求集合b\_Set和集合c\_Set的交集，并集，差集和对称差集

import numpy as np  
  
# 创建空集合 a\_Set  
a\_Set = set()  
  
# 创建一个集合 b\_Set 含有 1000 个元素，每个元素是 0-10000 之间的一个随机数  
b\_Set = set(np.random.randint(0, 10001, 1000))  
  
# 创建一个集合 c\_Set 含有 1000 个元素，每个元素是 8000-20000 之间的一个随机数  
c\_Set = set(np.random.randint(8000, 20001, 1000))  
  
intersection = b\_Set & c\_Set # 交集  
union = b\_Set | c\_Set # 并集  
difference = b\_Set - c\_Set # 差集  
symmetric\_difference = b\_Set ^ c\_Set # 对称差集  
  
print("交集:", intersection)  
print("并集:", union)  
print("差集:", difference)  
print("对称差集:", symmetric\_difference)

**结果：**



1. **Numpy练习，有关鸢尾花数据处理（至少选择8题）**

**数据见附件iris.csv（SepalLength:花萼长度 SepalWidth：花萼宽度 PetalLength:花瓣长度 PetalWidth:花瓣宽度）**

1. 导入鸢尾属植物数据集，保持文本不变。

import pandas as pd  
a = pd.read\_csv('Iris.csv')  
print(a)

**结果：**

图形用户界面

描述已自动生成

2. 求出鸢尾属植物萼片长度的平均值、中位数和标准差。

import pandas as pd  
a = pd.read\_csv('Iris.csv')  
s = a.iloc[:,1]  
  
  
mean = s.mean() #平均值  
median = s.median() #中位数  
stddev = s.std() #标准差  
  
print("平均值：", mean)  
print("中位数：", median)  
print("标准差：", stddev)

**结果：**

文本

描述已自动生成

3. 创建一种标准化形式的鸢尾属植物萼片长度，其值正好介于0和1之间，这样最小值为0，最大值为1

import pandas as pd  
import numpy as np  
iris=pd.read\_csv('iris.csv',index\_col=0)  
  
len\_min = min(iris.iloc[:, 0])  
len\_max = max(iris.iloc[:, 0])  
standardized\_length = (iris.iloc[:, 0]-len\_min)/(len\_max-len\_min)  
print(standardized\_length)

**结果：**

文本

描述已自动生成

4. 找到鸢尾属植物**萼片长度**的第25和第75百分位数

import pandas as pd  
import numpy as np  
a = pd.read\_csv('Iris.csv')  
percentile = np.percentile(a.iloc[:,1],[25,75])  
print(percentile)

**结果：**

文本

描述已自动生成

5. 把iris\_data数据集中的10个随机位置修改为np.nan值。

import pandas as pd  
import numpy as np  
a = pd.read\_csv('Iris.csv')  
np.random.seed(20201201)  
line = np.random.randint(low=0, high=a.shape[0], size=10)  
col = np.random.randint(low=0, high=a.shape[1]-1, size=10)  
for i in range(10):  
 a.iloc[line[i], col[i]] = np.nan  
print(a[0:10])

**结果：**

图形用户界面

描述已自动生成

18. 在鸢尾花数据集的petalwidth（第5列）中查找第一次出现的值大于1.0的位置。

import pandas as pd  
import numpy as np  
a = pd.read\_csv('Iris.csv')  
index = np.where(a.iloc[:, 4] > 1.0)  
print(index)  
print(index[0][0])

**结果：**

电脑的屏幕截图

描述已自动生成

10. 找出iris\_data是否有任何缺失值。

import pandas as pd  
import numpy as np  
a = pd.read\_csv('Iris.csv')  
print(a[np.sum(a.isna(), axis=1) >= 1])

**结果：**

文本

描述已自动生成

7. 筛选具有SepalWidth（第3列）< 4.0并且PetalLength（第4列）> 1.6的iris\_data行。

import pandas as pd  
import numpy as np  
a = pd.read\_csv('Iris.csv')  
print(a[(a.iloc[:,1]<5.0) & (a.iloc[:,3]>1.5)])

**结果：**

图形用户界面

中度可信度描述已自动生成