

《人工智能编程语言》第一次作业

陈海弘 23354049

2024.10.12

目录

1	作业	3
1.1	第一题	3
1.2	第二题	4
1.3	第三题	5
1.4	第四题	6
2	总结	7

1 作业

1.1 第一题

第一题 (20 分) 请编写一个 Python 程序, 该程序要能接受用户输入的一个任意长度的整数列表 (先输入列表长度, 再输入对应的数值), 并执行以下操作: 1. 将列表中的所有奇数元素提取出来, 存放在一个新的元组中, 并将结果打印出来。2. 计算列表中所有偶数的平均值, 并将结果打印出来。

思绪: 程序要求输入一个整数列表, 首先要求输入列表长度, 然后输入对应的数值。接着将列表中的奇数元素提取出来, 存放在一个新的元组中, 并将结果打印出来。最后计算列表中所有偶数的平均值, 并将结果打印出来。先用 `input()` 函数得到列表的长度, `for` 循环获得元素。奇数和偶数的提取可以同时进行。因为元组不可变, 先创立一个空列表, 然后用 `append()` 函数将奇数元素添加到列表中, 最后用 `tuple()` 函数将列表转换为元组。计算偶数的平均值, 用一个变量存储偶数的和, 一个变量存储偶数的个数, 最后用和除以个数得到平均值。

代码程序实现:

```
6 lens=int(input())
7 lis=[]
8 for i in range(lens): #使用循环输入列表
9     lis.append(int(input()))
10
11 odd = [] #创建一个空列表
12 even=0 #记录偶数的和
13 evens=0 #记录偶数的个数
14
15 for x in lis:
16     if x%2==1:
17         odd.append(x) #将奇数添加到odd列表中
18     else:
19         even+=x #将偶数相加
20         evens+=1 #记录偶数的个数
21 odd=tuple(odd) #将odd列表转换为元组
22
23 print(odd)
24 print(even/evens)
```

图 1: code1

```
/Users/chenhaihong/Pyth
6
1
2
3
4
5
6
(1, 3, 5)
4.0
```

图 2: terminal1

1.2 第二题

第二题 (25 分) 请编写一个 Python 程序, 该程序接受一个字符串'Hello world!'作为输入, 并执行以下操作: 1. 统计字符串中每个字符出现的次数, 并将结果存储在一个字典中, 其中键是字符, 值是字符的出现次数, 打印出字典中每个字符及其出现次数,。2. 创建一个包含字符串中唯一字符的集合, 并打印这个集合。3. 将字符串中所有字母变为大写。4. 利用字符串的切片操作提取子串, 提取字符串前 5 个字符、后 5 个字符、偶数索引位置的字符, 并将他们分别输出。5. 将字符串中的空格替换为",", 然后以",," 作为间隔将字符串分成两段保存到列表中, 再分别输出。

思绪: 利用 for 循环和字典的键值对, 统计字符串中每个字符出现的次数, 并将结果存储在一个字典中。创建一个空集合, 利用 for 循环将字符串中的唯一字符添加到集合中。利用 upper() 函数将字符串中所有字母变为大写。利用切片操作提取子串, 提取字符串前 5 个字符、后 5 个字符、偶数索引位置的字符。利用 replace() 函数将字符串中的空格替换为",", 然后以",," 作为间隔将字符串分成两段保存到列表中。

代码程序实现:

```
20 str=input()
21 dic={}
22 for char in str:
23     if char in dic: #如果字典中已经有这个字符, 就将这个字符的值加1
24         dic[char]+=1
25     else: #如果字典中没有这个字符, 就将这个字符的值设为1
26         dic[char]=1
27 print(dic)
28
29 only=set()
30 for char in str:
31     if dic[char]==1:
32         only.add(char) #将只出现一次的字符添加到集合中
33 print(only)
34
35 upper=str.upper() #将字符串中的字母全部变为大写
36 print(upper)
37
38 print(str[:5]) #输出前5个字符
39 print(str[-5:]) #输出后5个字符
40 print(str[::2]) #输出偶数索引位置的字符
41
42 strnew=str.replace(' ','') #将字符串中的空格替换为""
43 lis=strnew.split(',') #以","作为间隔将字符串分成两段保存到列表中
44 print(lis)
```

图 3: code2

```
/Users/chenhaihong/PycharmProjects/pythonProject/.venv/bin/python /Users/c
Hello world!
{'H': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 1, '!': 1}
{'H', 'e', 'd', 'l', 'o', 'r', 'w'}
HELLO WORLD!
Hello
orld!
!lowrd
['Hello', 'world!']
```

图 4: terminal2

1.3 第三题

第三题（25 分） 请编写一个 python 程序，该程序接受学生成绩信息序列作为输入，输出成绩从高到低、从低到高、按某一门成绩的排列。相同成绩按先录入排列在前的规则处理。数据如下：（数据规则：学生姓名高数成绩英语成绩大物成绩） SanZhang 70 80 61 SiLi 86 77 81 WuWang 88 90 77 MingLi 60 77 81 MiWang 71 70 60 HaiLi 88 78 89 HeWang 70 90 80 LiWang 67 71 70 要求：利用字典容器，完成对于每个学生成绩数据的储存，并将所有学生储存于列表中；在此基础上，按照总分从高到低的学生名单，总分从低到高的学生名单，三门课成绩从高到低的学生名单。给出具体的排序结果、涉及的代码，以及简要介绍自己设计的排序算法原理

思绪：学生成绩是输入，将学生成绩用 for 循环遍历后先储存单个作为字典，再将字典作为元素添加到列表中。利用字典容器，完成对于每个学生成绩数据的储存，并将所有学生储存于列表中。在此基础上，用 sort 函数按照总分从高到低的学生名单，总分从低到高的学生名单，三门课成绩从高到低的学生名单。

重要代码、程序实现：

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# 第三题
# 请编写一个 python 程序，该程序接受学生成绩信息序列作为输入，输出成绩从高到低、从低到高、按某一门成绩的排列。相同成绩按先录入排列在前的规则处理。数据如下：（数据规则：学生姓名高数成绩英语成绩大物成绩） SanZhang 70 80 61 SiLi 86 77 81 WuWang 88 90 77 MingLi 60 77 81 MiWang 71 70 60 HaiLi 88 78 89 HeWang 70 90 80 LiWang 67 71 70
# 要求：利用字典容器，完成对于每个学生成绩数据的储存，并将所有学生储存于列表中；在此基础上，按照总分从高到低的学生名单，总分从低到高的学生名单，三门课成绩从高到低的学生名单。给出具体的排序结果、涉及的代码，以及简要介绍自己设计的排序算法原理

# 定义学生成绩数据
students = [
    {'name': 'SanZhang', 'math': 70, 'english': 80, 'physics': 61, 'total': 211},
    {'name': 'SiLi', 'math': 86, 'english': 77, 'physics': 81, 'total': 244},
    {'name': 'WuWang', 'math': 88, 'english': 90, 'physics': 77, 'total': 255},
    {'name': 'MingLi', 'math': 60, 'english': 77, 'physics': 81, 'total': 218},
    {'name': 'MiWang', 'math': 71, 'english': 70, 'physics': 60, 'total': 201},
    {'name': 'HaiLi', 'math': 88, 'english': 78, 'physics': 89, 'total': 255},
    {'name': 'HeWang', 'math': 70, 'english': 90, 'physics': 80, 'total': 240},
    {'name': 'LiWang', 'math': 67, 'english': 71, 'physics': 70, 'total': 208}
]

# 按照总分从高到低排序
students.sort(key=lambda x: x['total'], reverse=True)

# 按照总分从低到高排序
students.sort(key=lambda x: x['total'])

# 按照数学成绩从高到低排序
students.sort(key=lambda x: x['math'], reverse=True)

# 按照英语成绩从高到低排序
students.sort(key=lambda x: x['english'], reverse=True)

# 按照物理成绩从高到低排序
students.sort(key=lambda x: x['physics'], reverse=True)
```

图 5: code3

```
按照总分从高到低的学生名单:
[{'name': 'WuWang', 'math': 88, 'english': 90, 'physics': 77, 'total': 255},
 {'name': 'HaiLi', 'math': 88, 'english': 78, 'physics': 89, 'total': 255},
 {'name': 'SiLi', 'math': 86, 'english': 77, 'physics': 81, 'total': 244},
 {'name': 'HeWang', 'math': 70, 'english': 90, 'physics': 80, 'total': 240},
 {'name': 'MingLi', 'math': 60, 'english': 77, 'physics': 81, 'total': 218},
 {'name': 'SanZhang', 'math': 70, 'english': 80, 'physics': 61, 'total': 211},
 {'name': 'LiWang', 'math': 67, 'english': 71, 'physics': 70, 'total': 208},
 {'name': 'MiWang', 'math': 71, 'english': 70, 'physics': 60, 'total': 201}]

按照总分从低到高的学生名单:
[{'name': 'MiWang', 'math': 71, 'english': 70, 'physics': 60, 'total': 201},
 {'name': 'LiWang', 'math': 67, 'english': 71, 'physics': 70, 'total': 208},
 {'name': 'MingLi', 'math': 60, 'english': 77, 'physics': 81, 'total': 218},
 {'name': 'SanZhang', 'math': 70, 'english': 80, 'physics': 61, 'total': 211},
 {'name': 'WuWang', 'math': 88, 'english': 90, 'physics': 77, 'total': 255},
 {'name': 'HaiLi', 'math': 88, 'english': 78, 'physics': 89, 'total': 255}]

按照数学成绩从高到低的学生名单:
[{'name': 'WuWang', 'math': 88, 'english': 90, 'physics': 77, 'total': 255},
 {'name': 'HaiLi', 'math': 88, 'english': 78, 'physics': 89, 'total': 255},
 {'name': 'SiLi', 'math': 86, 'english': 77, 'physics': 81, 'total': 244},
 {'name': 'HeWang', 'math': 70, 'english': 90, 'physics': 80, 'total': 240},
 {'name': 'MingLi', 'math': 60, 'english': 77, 'physics': 81, 'total': 218},
 {'name': 'SanZhang', 'math': 70, 'english': 80, 'physics': 61, 'total': 211},
 {'name': 'LiWang', 'math': 67, 'english': 71, 'physics': 70, 'total': 208},
 {'name': 'MiWang', 'math': 71, 'english': 70, 'physics': 60, 'total': 201}]

按照英语成绩从高到低的学生名单:
[{'name': 'WuWang', 'math': 88, 'english': 90, 'physics': 77, 'total': 255},
 {'name': 'HeWang', 'math': 70, 'english': 90, 'physics': 80, 'total': 240},
 {'name': 'SanZhang', 'math': 70, 'english': 80, 'physics': 61, 'total': 211},
 {'name': 'HaiLi', 'math': 88, 'english': 78, 'physics': 89, 'total': 255},
 {'name': 'SiLi', 'math': 86, 'english': 77, 'physics': 81, 'total': 244},
 {'name': 'MingLi', 'math': 60, 'english': 77, 'physics': 81, 'total': 218},
 {'name': 'LiWang', 'math': 67, 'english': 71, 'physics': 70, 'total': 208},
 {'name': 'MiWang', 'math': 71, 'english': 70, 'physics': 60, 'total': 201}]

按照物理成绩从高到低的学生名单:
[{'name': 'HaiLi', 'math': 88, 'english': 78, 'physics': 89, 'total': 255},
 {'name': 'SiLi', 'math': 86, 'english': 77, 'physics': 81, 'total': 244},
 {'name': 'MingLi', 'math': 60, 'english': 77, 'physics': 81, 'total': 218},
 {'name': 'WuWang', 'math': 88, 'english': 90, 'physics': 77, 'total': 255},
 {'name': 'HeWang', 'math': 70, 'english': 90, 'physics': 80, 'total': 240},
 {'name': 'LiWang', 'math': 67, 'english': 71, 'physics': 70, 'total': 208},
 {'name': 'SanZhang', 'math': 70, 'english': 80, 'physics': 61, 'total': 211},
 {'name': 'MiWang', 'math': 71, 'english': 70, 'physics': 60, 'total': 201}]
```

图 6: terminal3

1.4 第四题

第四题（30 分）请编写一个 Python 程序，要求该程序可以帮助用户管理他们的购物清单。用户可以添加、查看、删除和编辑购物清单中的商品，还可以查看购物清单的总价格。各功能形成菜单供用户选择，包括退出功能。提示：购物清单是一个字典，其中商品名称是键，商品价格是值。通过 input 来输入每次要执行的操作，用循环来实现多次执行操作。

思绪：程序要求实现添加，查看，删除，编辑，查看总价格等功能，首先创建了菜单显示函数，这个函数可以显示如上功能。然后定义了其他的函数，添加商品函数 add-item，查看商品函数 view-item，删除函数等等。要调用这个函数，需要在终端调用 main 函数，这个函数初始化一个购物清单字典，用 while 不断循环，直到用户输入终止代码才停止。然后通过用户的输入，不断调用前面的函数来实现功能。

重要代码实现、终端展示：

```
def display_menu(): #显示菜单
    print("1. Add item")
    print("2. View item")
    print("3. Delete item")
    print("4. Edit item")
    print("5. View total price")
    print("6. Exit")
    choice = input("Enter your choice: ")
    return choice

def add_item(shopping_list): #添加商品
    item = input("Enter item name: ") #输入商品名称
    name, price = item.split()
    shopping_list[name] = float(price) #利用创建的字典shopping_list，将商品名称作为键，商品价格作为值，添加到字典中

def view_item(shopping_list): #查看商品
    for name, price in shopping_list.items(): #遍历字典shopping_list，输出商品名称和价格
        print(name, price)

def delete_item(shopping_list): #删除商品
    item = input("Enter item name: ")
    if item in shopping_list: #如果商品在字典shopping_list中
        del shopping_list[item] #删除商品
    else:
        print("Item not found")

def edit_item(shopping_list): #编辑商品
    item = input("Enter item name and price separated by a space: ")
    name, price = item.split()
    position = input("Enter 'beginning' to add to the beginning or 'end' to add to the end: ").strip().lower()
    if position == 'beginning': #如果输入的位置是'beginning'
        shopping_list.insert(0, (name, float(price))) #在字典shopping_list的第一个位置添加商品，也就是在字典的开头添加商品
    elif position == 'end':
        shopping_list.append((name, float(price))) #在字典shopping_list的最后一个位置添加商品，也就是在字典的末尾添加商品
    else:
        print("Invalid position. Item not added.")

def view_total_price(shopping_list):
    total_price = sum(shopping_list.values()) #通过sum函数计算字典shopping_list中所有商品价格的总和
    print("Total price: ", total_price)
```

图 7: code4

```

Enter your choice: 2
happy 2000.0
sad 9999.0
1. Add item
2. View item
3. Delete item
4. Edit item
5. View total price
6. Exit
Enter your choice: 3
Enter item name: sad
1. Add item
2. View item
3. Delete item
4. Edit item
5. View total price
6. Exit
Enter your choice: 2
happy 2000.0
1. Add item
2. View item
3. Delete item
4. Edit item
5. View total price
6. Exit

```

图 8: terminal4

```

Enter your choice: 5
Total price: 2200.0
1. Add item
2. View item
3. Delete item
4. Edit item
5. View total price
6. Exit
Enter your choice: 6
>>> 

```

图 9: terminal4

2 总结

以上四个作业，分别是对 Python 基础知识的综合运用，包括列表、元组、字典等数据结构的使用，以及对字符串的操作，对函数的定义和调用，对循环的使用等等。有些细枝末节的东西需要注意比如 input 函数输入的是字符串，需要转换为整数，对字典，列表，元组，集合的定义，操作也很容易遗忘。刚学的循环语句 for，结构是 for i in ，后面可以接列表，元组，集合，字典，字符串等等。if 语句中一定要记得缩进，这里和 c 语言不同，缩进是 python 的一个特点。总的来说，这次作业是对 Python 基础知识的一个综合运用，我对 Python 的基础知识有了更深的理解，对 Python 的编程能力有了更深的认识。