Python程序设计——第二次实验报告

2311095 宋卓伦 计算机科学与技术 2024.09.13

一、实验目的

编写选择结构和循环结构的程序，掌握内置数据类型的基本使用方法，熟练运用各类运算符及if、for、while解决实际问题，具备通过调试发现并解决程序逻辑问题的能力。

二、实验描述

编写程序，使用选择排序算法将列表中的元素按照升序方式排列。

假设列表中有n个元素，则选择排序算法处理过程：

（1）从n个元素中找出具有最小值的元素，如果其不是第1个元素则将其与第1个元素交换。

（2）从后n-1个元素中找出具有最小值的元素，如果其不是第2个元素则将其与第2个元素交换。

……

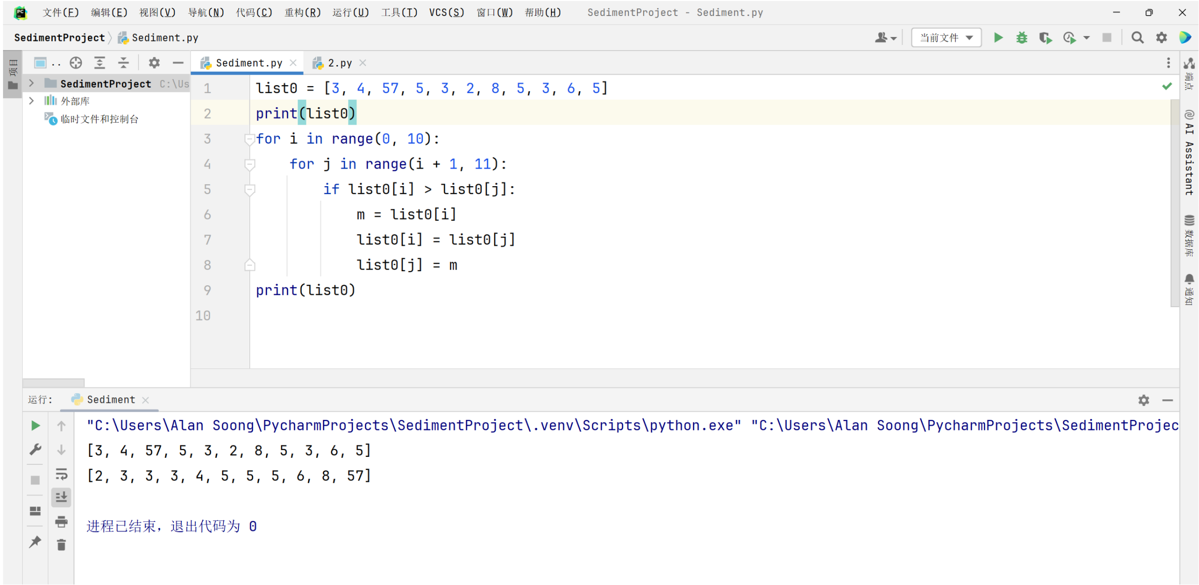
（i）从后n-i+1个元素中找出具有最小值的元素，如果其不是第i个元素，则将其与第i个元素交换。

……

（n-1）从后2个元素中找出具有最小值的元素，如果其不是第n-1个元素则将其与第n-1个元素交换。

三、实验代码

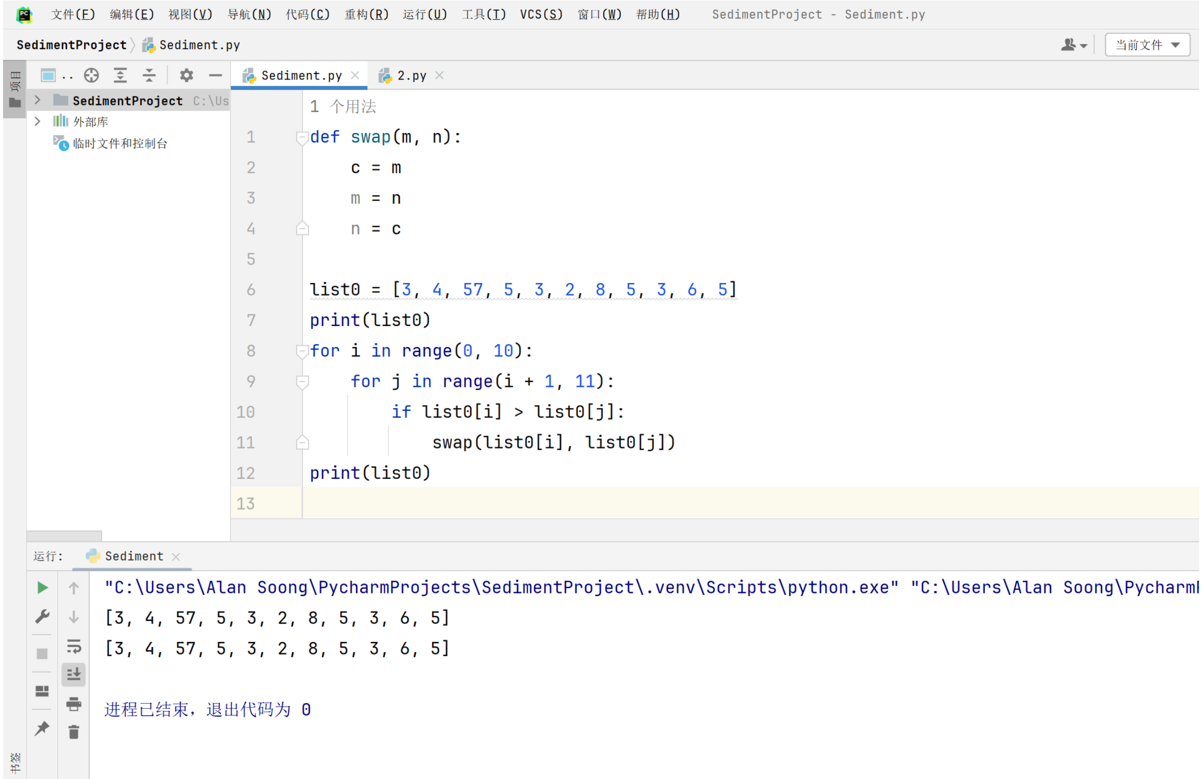
```python
list0 = [3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
print(list0)


for i in range(0,10):
    for j in range(i + 1,11):
        if list0[i]> list0[j]:
            m = list0[i]
            list0[i] = list0[j]
            list0[j] = m


print(list0)

```


文件(F) 编辑(E) 视图(V) 导航(N) 代码(C) 重构(R) 运行(U) 工具(T) VCS(S) 窗口(W) 帮助(H)
SedimentProject - Sediment.py
SedimentProject 〉 Sediment.py
当前文件
Sediment.py
2.py
SedimentProject C:\Us
外部库
临时文件和控制台
运行: Sediment

```python
list0 = [3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
print(list0)
for i in range(0, 10):
    for j in range(i + 1, 11):
        if list0[i] > list0[j]:
            m = list0[i]
            list0[i] = list0[j]
            list0[j] = m
print(list0)
```

```
"C:\Users\Alan Soong\PycharmProjects\SedimentProject\.venv\Scripts\python.exe" "C:\Users\Alan Soong\PycharmProjects\SedimentProjec
[3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
[2, 3, 3, 3, 4, 5, 5, 5, 6, 8, 57]

进程已结束，退出代码为 0
```

通过两个循环实现了List的排序，注意到如果命名swap函数的话，方法不当会导致数据无法正确实现交换从而导致列表元素未发生改变。


文件(F) 编辑(E) 视图(V) 导航(N) 代码(C) 重构(R) 运行(U) 工具(T) VCS(S) 窗口(W) 帮助(H)
SedimentProject - Sediment.py
SedimentProject 〉 Sediment.py
当前文件
Sediment.py
2.py
SedimentProject C:\Us
外部库
临时文件和控制台
1 个用法
运行: Sediment

```python
def swap(m, n):
    c = m
    m = n
    n = c

list0 = [3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
print(list0)
for i in range(0, 10):
    for j in range(i + 1, 11):
        if list0[i] > list0[j]:
            swap(list0[i], list0[j])
print(list0)
```

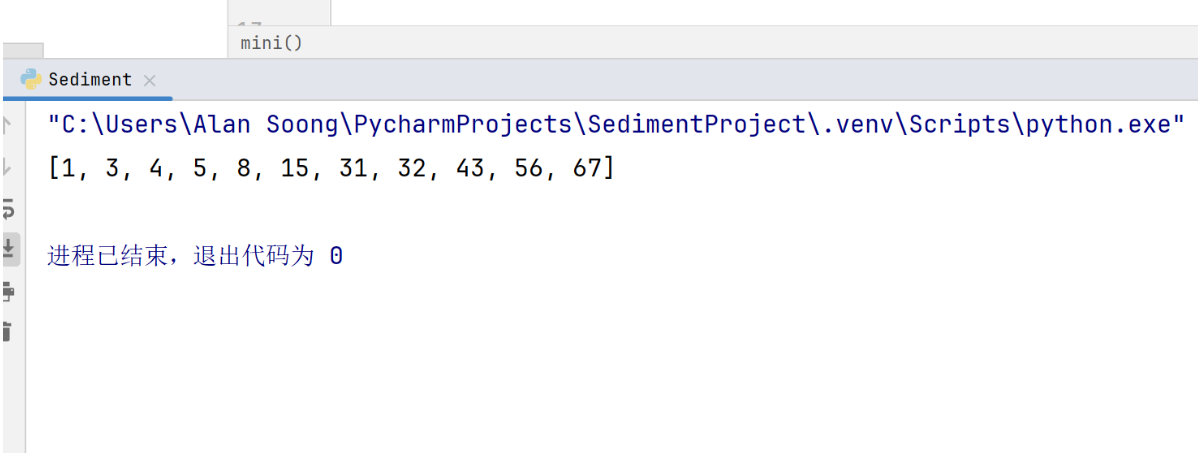
```
"C:\Users\Alan Soong\PycharmProjects\SedimentProject\.venv\Scripts\python.exe" "C:\Users\Alan Soong\Pycharml
[3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
[3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]

进程已结束，退出代码为 0
```

补充：如果自定义列表，如何进行排序?

我们先引入一个已知列表：

```python
# -*- coding: utf-8 -*-
list0 = [1, 4, 5, 31, 67, 15, 3, 8, 56, 43, 32]


# 查找最小元素值的下标;
def mini(a, b, list0):
    m = a
    for i in range(a + 1, b + 1):
        if list0[m] > list0[i]:
            m = i
    return m
```

```python


for i in range(0, 10):
    k = mini(i + 1, 10, list0)
    if list0[i] > list0[k]:
        list0[i], list0[k] = list0[k], list0[i]


print(list0)

```

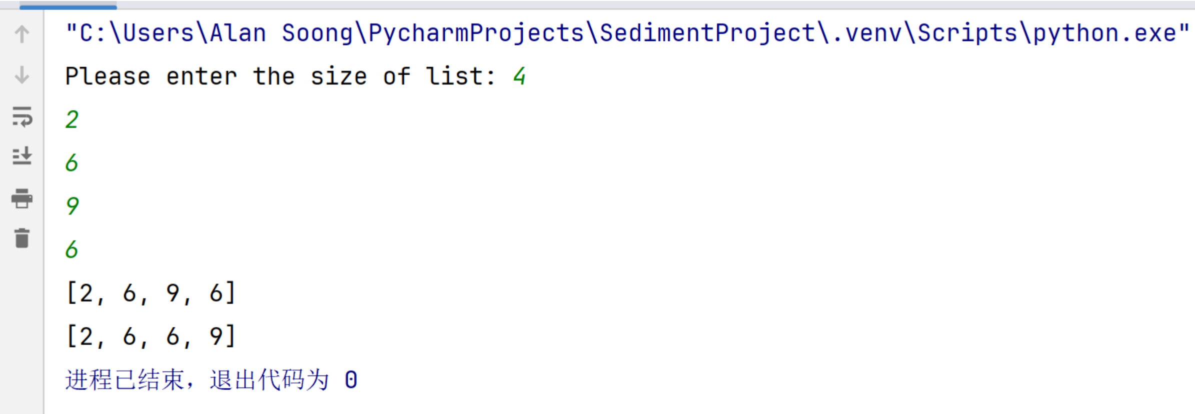


```
"C:\Users\Alan Soong\PycharmProjects\SedimentProject\.venv\Scripts\python.exe"
[1, 3, 4, 5, 8, 15, 31, 32, 43, 56, 67]

进程已结束，退出代码为 0
```

上面的方法严格按照了要求的方法，通过区间的收缩寻找到了区间内的最小值并返回其下标。

接下来是需要输入的列表：

```python
def mini(a, b, list0):
    m = a
    for i in range(a + 1, b + 1):
        if list0[m] > list0[i]:
            m = i
    return m


n = int(input("Please enter the size of list: "))
list1 = []

for j in range(0, n):
    c = int(input())
    list1.append(c)

print(list1)

for i in range(0, n - 1):
    k = mini(i + 1, n - 1, list1)
    if list1[i] > list1[k]:
        list1[i], list1[k] = list1[k], list1[i]

print(list1, end="\r")

```



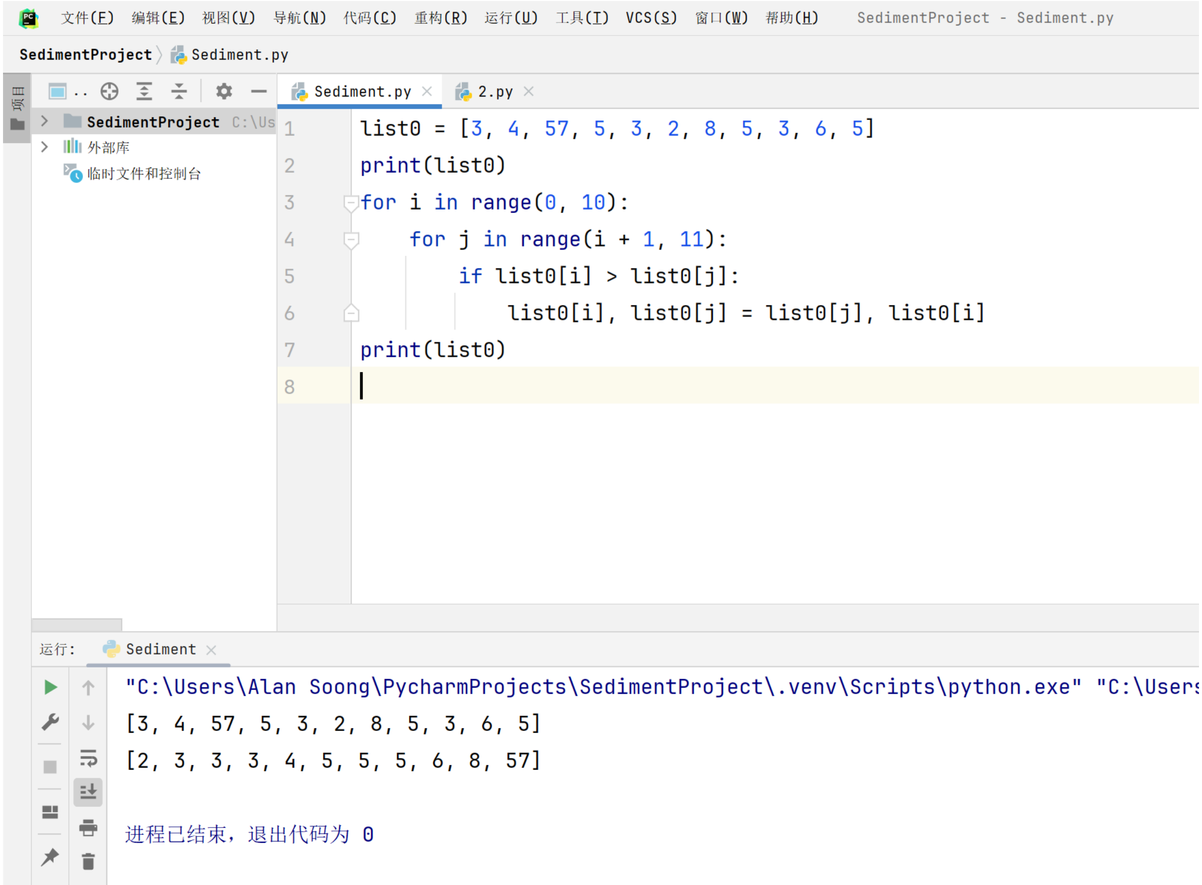
```
"C:\Users\Alan Soong\PycharmProjects\SedimentProject\.venv\Scripts\python.exe"
Please enter the size of list: 4
2
6
9
6
[2, 6, 9, 6]
[2, 6, 6, 9]
进程已结束，退出代码为 0
```

我们通过list自带的append函数完成了列表元素的插入，然后和上述方法一样实现排序。

四、实验反思

遇到的问题：定义交换函数无法正确交换两个数的值

解决过程：直接使用语句“y,x = x,y”即可



文件(F) 编辑(E) 视图(V) 导航(N) 代码(C) 重构(R) 运行(U) 工具(T) VCS(S) 窗口(W) 帮助(H)
SedimentProject - Sediment.py
SedimentProject
Sediment.py
项目
SedimentProject C:\Us
外部库
临时文件和控制台
Sediment.py
2.py
运行:
Sediment

```python
list0 = [3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
print(list0)
for i in range(0, 10):
    for j in range(i + 1, 11):
        if list0[i] > list0[j]:
            list0[i], list0[j] = list0[j], list0[i]
print(list0)
```

```
"C:\Users\Alan Soong\PycharmProjects\SedimentProject\.venv\Scripts\python.exe" "C:\Users
[3, 4, 57, 5, 3, 2, 8, 5, 3, 6, 5]
[2, 3, 3, 3, 4, 5, 5, 5, 6, 8, 57]

进程已结束，退出代码为 0
```

此外，还有一些基于实验代码写法的逻辑问题，比如“最著名”的“左闭右开区间”，（0，n）的区间实际上是指0~n-1的数值范围，这里多处地方已经修正。

五、补充的简易代码设计：基于while和选择语句针对GPA的计算器

代码示例：

```python
# -*- coding: utf-8 -*-
# Here is the coding of calculating GPA(score):

M = 0
S = 0

while True:
    print("Enter your datas: ")
    n = float(input("Here is your score: "))
    m = float(input("Here is your credit: "))
    M += m
    S += m * n
    s = S / M
    p = input("Go on? ")
    if p != 'q':
        continue
    break

if s < 1.0:
    print('Warning! Your GPA is low! Please work harder in the next semester!')
elif s >= 3.8:
    print('Excellent! Your GPA is quite high! Please keep going!')

print('Your GPA is', '%.4f' % s, end='\r')

```

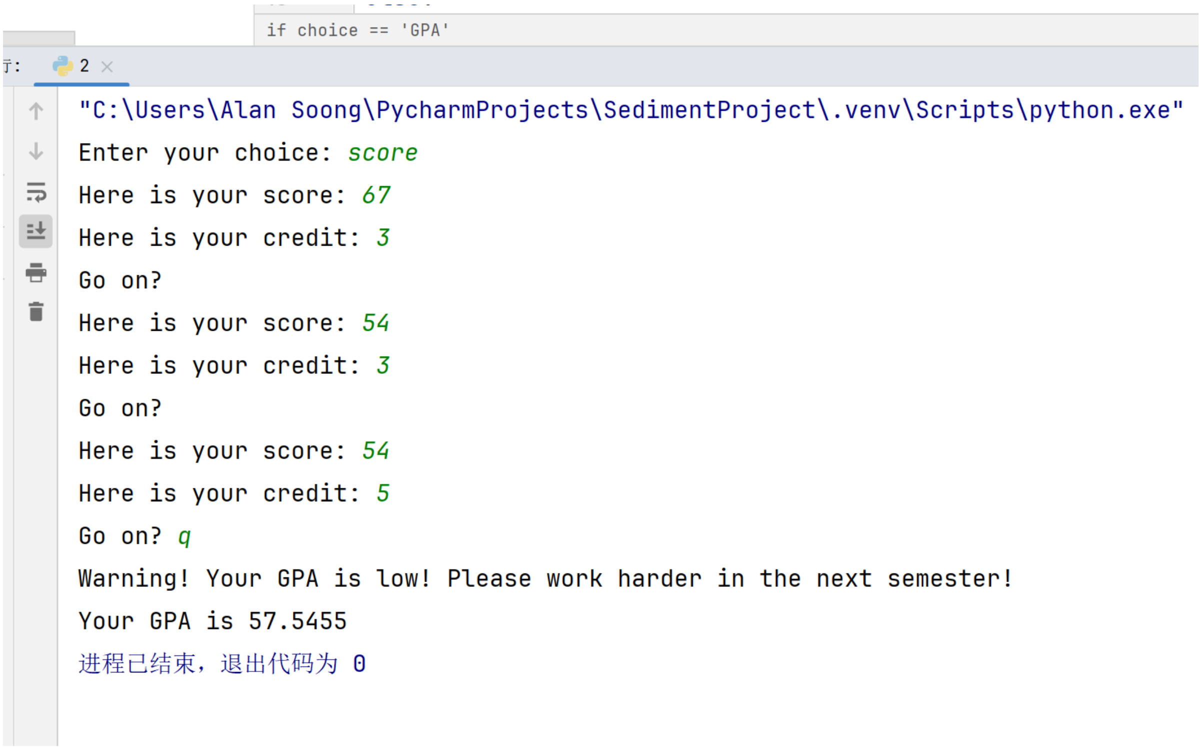
先定义了M、S两个中间变量，在计算中充当总学分绩点和总学分的作用，通过while和if、else两个功能实现了循环输入，并且可以自行选择是否退出，最后打印结果。同时根据GPA情况打印提示并输出平均学分绩点。控制在了4位有效小数上。

针对学分绩和学分绩点二者不同的情况，我继续进行如下改进：

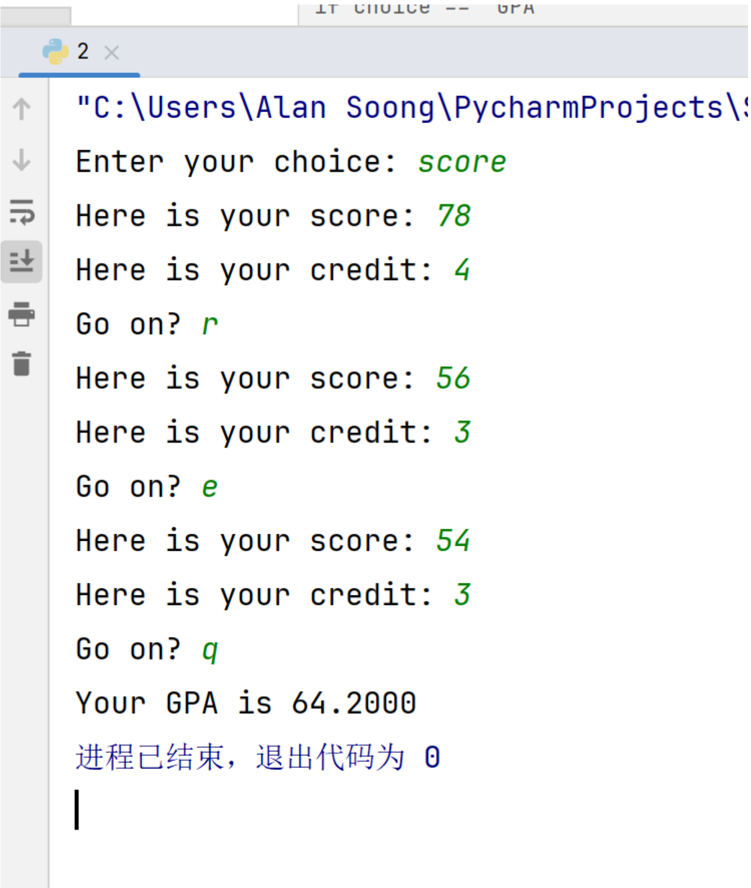
```python
def calculate():
    M = 0
    S = 0
    while True:
        m = float(input("Here is your credit: "))
        n = float(input("Here is your score: "))
        M += m
        S += m * n
        p = input("Go on? ")
        if p.lower() != 'q':
            continue
        else:
            break
    if M == 0:
        return 0
    return S/M


```

```python
# For p in calculate(), you‘d better input a letter, not pushing other bottoms.

def case1(s):
    if s < 1.0:
        print('Warning! Your GPA is low! Please work harder in the next semester!')

    elif s >= 3.8:
        print('Excellent! Your GPA is quite high! Please keep going!')


def case2(s):
    if s < 60:
        print('Warning! Your GPA is low! Please work harder in the next semester!')
    elif s >= 90:
        print('Excellent! Your GPA is quite high! Please keep going!')


choice = input("Enter your choice: ")

if choice == 'GPA':
    final = calculate()
    case1(final)
elif choice == 'score':
    final = calculate()
    case2(final)
else:
    print('Please restart and input it again! ')

print('Your GPA is', '%.4f' % final, end='\r')

```

这样通过函数+if、else语句实现了GPA和学分绩的切换。此外lower函数可以有效避免大小写问题。



```
"C:\Users\Alan Soong\PycharmProjects\SedimentProject\.venv\Scripts\python.exe"
Enter your choice: score
Here is your score: 67
Here is your credit: 3
Go on?
Here is your score: 54
Here is your credit: 3
Go on?
Here is your score: 54
Here is your credit: 5
Go on? q
Warning! Your GPA is low! Please work harder in the next semester!
Your GPA is 57.5455
进程已结束，退出代码为 0
```



```
"C:\Users\Alan Soong\PycharmProjects\
Enter your choice: score
Here is your score: 78
Here is your credit: 4
Go on? r
Here is your score: 56
Here is your credit: 3
Go on? e
Here is your score: 54
Here is your credit: 3
Go on? q
Your GPA is 64.2000
进程已结束，退出代码为 0
```

注意到由于引用过多无效形参会导致数据无法正确显示登问题，我做完了上述改进。