problem set 1 实验报告

16337183 孟衍璋

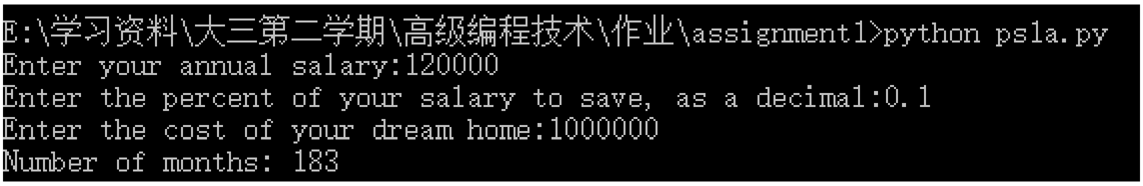
ps1a

第一道题只需要按月将收入累加，直到总共的钱大于首付。每个月的收入一共由三部分组成：现在已有的钱+每个月存下的月薪+投资所得的钱。

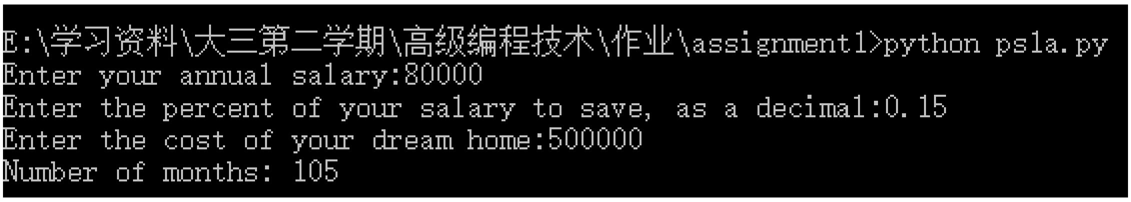
实现代码如下：

```
annual_salary = int(input("Enter your annual salary:"))
portion_saved = float(input("Enter the percent of your salary to save, as a decimal:"))
total_cost = int(input("Enter the cost of your dream home:"))

portion_down_payment = 0.25
current_savings = 0
r = 0.04

months = 0
# print(total_cost * portion_down_payment)
while(current_savings < total_cost * portion_down_payment):
    money = current_savings + annual_salary / 12 * portion_saved + current_savings * r / 12
    current_savings = money
    months += 1

print("Number of months:", months)
```

运行结果：



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1a.py
Enter your annual salary:120000
Enter the percent of your salary to save, as a decimal:0.1
Enter the cost of your dream home:1000000
Number of months: 183
```



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1a.py
Enter your annual salary:80000
Enter the percent of your salary to save, as a decimal:0.15
Enter the cost of your dream home:500000
Number of months: 105
```

ps1b

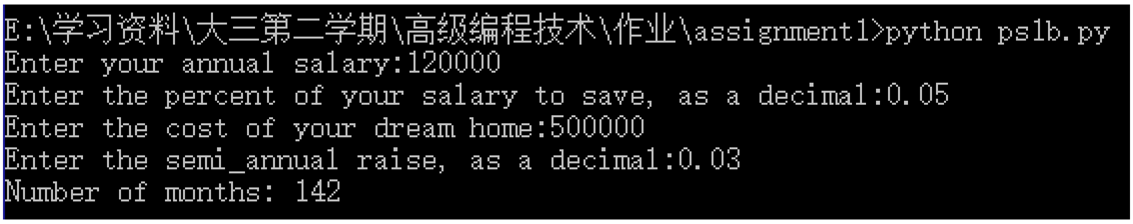
第二道题添加了年薪会随着工作时间的增长而增加的条件，于是在实现过程中，每六个月将年薪增加相应数额之后再计算。

实现代码如下：

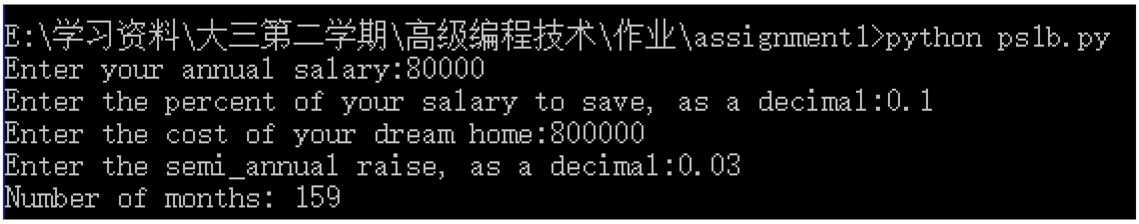
```
annual_salary = int(input("Enter your annual salary:"))
portion_saved = float(input("Enter the percent of your salary to save, as a decimal:"))
total_cost = int(input("Enter the cost of your dream home:"))
```

```
semi_annual_raise = float(input("Enter the semi_annual raise, as a decimal:"))

portion_down_payment = 0.25
current_savings = 0
r = 0.04

months = 0
# print(total_cost * portion_down_payment)
while(current_savings < total_cost * portion_down_payment):
    money = current_savings + annual_salary / 12 * portion_saved + current_savings * r / 12
    current_savings = money
    months += 1
    if months % 6 == 0:
        annual_salary = (semi_annual_raise + 1) * annual_salary

print("Number of months:", months)
```

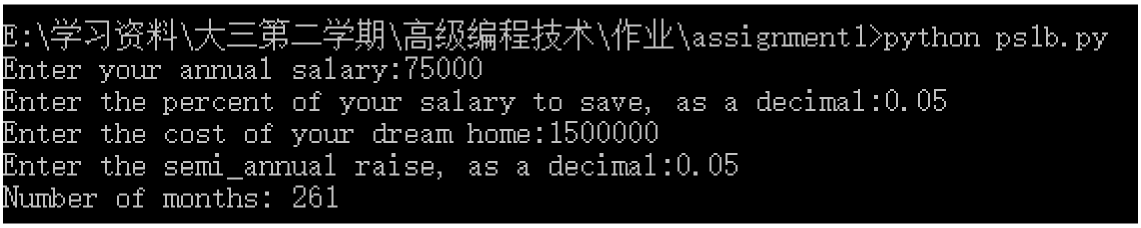
运行结果:



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1b.py
Enter your annual salary:120000
Enter the percent of your salary to save, as a decimal:0.05
Enter the cost of your dream home:500000
Enter the semi_annual raise, as a decimal:0.03
Number of months: 142
```



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1b.py
Enter your annual salary:80000
Enter the percent of your salary to save, as a decimal:0.1
Enter the cost of your dream home:800000
Enter the semi_annual raise, as a decimal:0.03
Number of months: 159
```



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1b.py
Enter your annual salary:75000
Enter the percent of your salary to save, as a decimal:0.05
Enter the cost of your dream home:1500000
Enter the semi_annual raise, as a decimal:0.05
Number of months: 261
```

ps1c

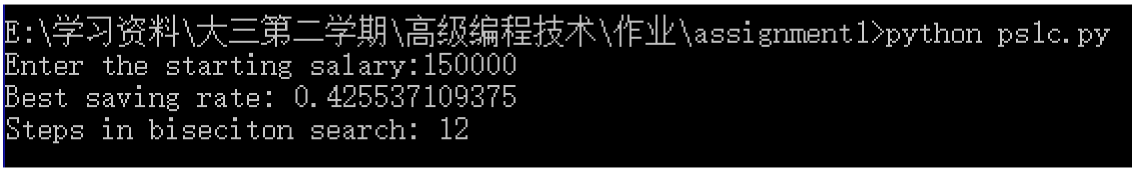
第三道题是除了初始年薪和每月存储的百分比之外确定了其他信息，输入一个初始年薪，计算每个月需要存储多少百分比的钱，才能保证在36个月以内筹齐指定价格的房子的首付。由于存储的百分比的值是在0-1之间，所以采取二分法，通过判断3年后存储的金额大小，来不断更新上下界，最后直到3年内筹集的钱刚好在所需要的钱正负100的误差范围之内。

实现代码如下:

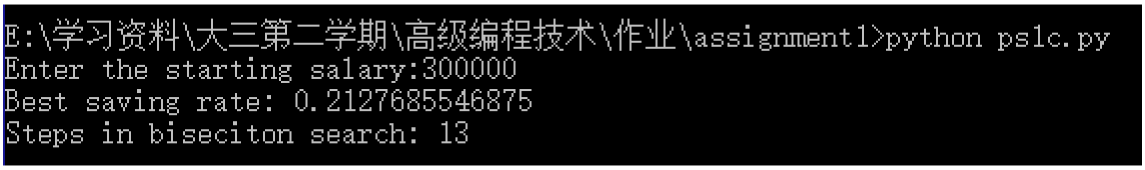
```
annual_salary = int(input("Enter the starting salary:"))

semi_annual_raise = 0.07
portion_down_payment = 0.25
current_savings = 0
total_cost = 1000000
r = 0.04
```

```
def money_after_36months(annual_salary, portion_saved):
    current_savings = 0
    months = 0
    while months <= 36:
        money = current_savings + annual_salary / 12 * portion_saved + current_savings * r /
12
        current_savings = money
        months += 1
        if months % 6 == 0:
            annual_salary = (semi_annual_raise + 1) * annual_salary
    return current_savings

def lowest_money(annual_salary):
    current_savings = 0
    months = 0
    portion_saved = 1
    while months <= 36:
        money = current_savings + annual_salary / 12 * portion_saved + current_savings * r /
12
        current_savings = money
        months += 1
        if months % 6 == 0:
            annual_salary = (semi_annual_raise + 1) * annual_salary
    return current_savings

if lowest_money(annual_salary) < total_cost * portion_down_payment:
    print("It is not possible to pay the down payment in three years.")

else:
    up = 1
    down = 0
    biseciton = 0
    while (current_savings <= portion_down_payment * total_cost + 100 and current_savings >=
portion_down_payment * total_cost - 100) == 0:
        portion_saved = 0.5 * (up + down)
        current_savings = money_after_36months(annual_salary, portion_saved)
        if current_savings >= portion_down_payment * total_cost + 100:
            up = portion_saved
        elif current_savings <= portion_down_payment * total_cost - 100:
            down = portion_saved
        biseciton += 1
        # print(biseciton, current_savings)

    print("Best saving rate:", portion_saved)
    print("Steps in biseciton search:", biseciton)
```

运行结果：



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1c.py
Enter the starting salary:150000
Best saving rate: 0.425537109375
Steps in biseciton search: 12
```



```
E:\学习资料\大三第二学期\高级编程技术\作业\assignment1>python ps1c.py
Enter the starting salary:300000
Best saving rate: 0.2127685546875
Steps in biseciton search: 13
```