

关于 ps1a 部分:

【题意理解】

一个刚毕业的大学生有份工作，想要买房，准备开始存钱付房子的首付。重点：

1: 由用户输入房价 `total_cost`,其中首付 25%，那他的目标就应该是存够 `target=0.25*total_cost`

2: 由用户输入他的年薪 `annual_salary`,以及他的每月用于存款部分的百分比 `,portion_saved`,其中存款年利率已知为 4%，即 0.04 。那么月利率就是 $(4.0/12) \%$ ， 即 $1.0/300$ 。

【解题思路】

$s=1+1.0/300$:表示用户本金月增加的倍数。

用户第 1 个月拿到工资，存 1000 元本金进去，第 2 个月后拥有 $1000*s+1000$

第 3 个月后拥有 $(1000*s+1000)*s+1000$

.....

用户每个月的账户上余额将是 上一个月余额乘以 s ，再加新存的 1000

为了验证我的猜想与计算方式是否正确，我先粗略的写了段代码，并用题目所给案例进行验证：

案例一验证：月存款 1000，183 月后能存够 250000

```
money=1000 #第一个月后拥有本金
s=1+1.0000/300 #每月存款增加倍数
i=2
months=183
while i < months+1:
    temp=money*s
    money=temp+1000 #i 月后拥有资金
    i+=1;
print(money)
```

```
In [8]: runfile('C:/Users/86156/.sp:
Users/86156/.spyder-py3')
249737.15913791445
```

当 months 取 182 时，他的存款是

```
In [9]: runfile('C:/Users/86
Users/86156/.spyder-py3')
251569.61633504086
```

当 months 取 183 时，他的存款是

案例 2 验证：月存款 1000，105 月后能存够 125000

```
money=1000 #第一个月后拥有本金
s=1+1.00000/300 #每月存款增加倍数
i=2
months=105
while i < months+1:
    temp=money*s
    money=temp+1000 #i 月后拥有资金
    i+=1;
print(money)
```

In [11]: runfile('C:/Users/86156/.spyder-py3')
当 months 取 104 时, 他的存款是 124059.02026985459

In [12]: runfile('C:/Users/86156/.spyder-py3')
当 months 取 105 时, 他的存款将是 125472.55033742078

由此判断, 以上计算方式是正确的, 我开始按要求修改变量命名, 并将代码稍作调整和完善, 详情见文件 ps1a.py.

程序以及运行结果如下 (除所给案例外, 还测试了几个):

```
7 annual_salary=int(input("Enter your annual salary: "))
8 portion_saved=float(input("Enter the percent of your salary to save,as a decimal: "))
9 total_cost=int(input("Enter the cost of your dream home: "))
10
11 money=(annual_salary*portion_saved)/12 #每月存款金额
12 target=total_cost*0.25 #存款目标金额
13
14 current_saving=money #一个月后存款
15 r=0.04
16 s=1+r/12 #每月存款增加倍数
17 i=2 #从第2个月开始赚利息
18 while 1>0:
19     temp=current_saving*s
20     current_saving=temp+money #i 月后拥有存款
21     if current_saving<target: #没存够, 继续存
22         i=i+1
23         continue
24     else: #存够了, 跳出循环
25         months=i
26         break
27 print("Number of months: ",months) #输出结果
```

```
In [24]: runfile('C:/Users/86156/.spyder-py3/ps1a.py', wdir='C:/Users/86156/.spyder-py3')
Enter your annual salary: 120000
Enter the percent of your salary to save,as a decimal: 0.10
Enter the cost of your dream home: 1000000
Number of months: 183
```

```
In [25]: runfile('C:/Users/86156/.spyder-py3/ps1a.py', wdir='C:/Users/86156/.spyder-py3')
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
```

关于 ps1b 部分:

【题意理解】

员工的工资会随着工作时间增加而增加, 用户输入增加的系数 semi_annual_raise, 每过六个月, 员工的工资将会在原来工资基础上增加 semi_annual_raise 倍, 其他条件, 如房子首付比例不变, 存款年利率不变, 要求员工需要存多少个月才能存够房子首付。

【解题思路】

大体程序设计结构与 ps1a 不变, while 循环, 在更新当前存款的时候稍作改动

1、员工六个月后的工资是六个月前的 $(1 + \text{semi_annual_raise})$ 倍, 即每过 6 个月, 他去存的钱会增加为原来的 $(1 + \text{semi_annual_raise})$ 倍。只需在循环增加一个 if 判断即可, 当月数 $\%6 == 1$ 时, 去存的钱增加相应倍数即可, 现贴上关键地方代码块:

```
19 i=2 #从第2个月开始赚利息
20 while 1>0:
21     temp=current_saving*s
22     if i%6==1: #每过6个月, 存款增加
23         money*= (1+semi_annual_raise)
24     current_saving=temp+money #i 月后拥有存款
```

仅在原基础上增加了一个 if 判断，外加输入操作。案例运行结果如下：

```
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

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

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
```

关于 ps1c 部分：

【题意理解】

条件改变了，由用户给出年薪，其他条件（如房子首付、年利率）不变，期限 3 年，求每个月该员工至少要将工资拿去存的比例 rate 。

【解题思路】

在 ps1b 的基础之上，改动一下，将计算 3 年的存款写成一个函数，参数是 rate 也就是员工每个月的存款比例，显然这是一个单调递增的函数，因此用二分（当然题目也有提示），并且要求二分的次数，二分显然是对于整数而言的，而求出的比例要求保留 4 位小数，因此我将参数放大到 10000 倍，来进行二分，将 (0, 1]范围的比例转换成[1,10000]的区间进行二分。

```
15 def func(x):
16     money=annual_salary*(x/120000)           #每月存款金额
17     current_saving=money                     #一个月后存款
18     i=2
19     while i<37:
20         temp=current_saving*inc
21         if i%6==1:                             #每过6个月，工资增加
22             money*=s
23             current_saving=temp+money          #i月后拥有存款
24             i=i+1
25     return current_saving-target
```

题目即求满足 $\text{func}(\text{rate}) > 0$ 的最小 rate 值。关键代码如下：

```

27 left=0
28 right=10000
29
30 if func(right)<0:
31     print("It is not possible to pay the down payment in three years.")
32 else:
33     counter=0
34     while left<right:
35         mid=int((left+right)/2)
36         counter+=1
37         if func(mid)<0:
38             left=mid+1
39         if func(mid)>0:
40             right=mid-1
41     rate=left/10000
42     print("Best savings rate: ",rate)
43     print("Steps in bisection search: ",counter)

```

相关案例运行结果如下：

```

Enter the starting salary: 150000
Best savings rate: 0.4411
Steps in bisection search: 13

In [26]: runfile('C:/Users/86156/.spyder-py3/
pslc.py', wdir='C:/Users/86156/.spyder-py3')

Enter the starting salary: 300000
Best savings rate: 0.2206
Steps in bisection search: 12

In [27]: runfile('C:/Users/86156/.spyder-py3/
pslc.py', wdir='C:/Users/86156/.spyder-py3')

Enter the starting salary: 10000
It is not possible to pay the down payment in
three years.

```

二分次数与结果存在一点误差，应该是二分方法不同导致的。