Python脚本语言

第七周 Python案例精选

陈世峰 岭南师范学院信息工程学院
chenshifeng@lingnan.edu.cn/13234075348

内容概要

1 •课后作业

2 •经典例题

Good good study, day day up

Let us study hard and make progress every day!

例5.1 天天向上 v1.0

一年365天，以第1天的能力值为基数，记为1.0，当好好学习时能力值相比前一天提高1‰，当没有学习时由于遗忘等原因能力值相比前一天下降1‰。每天努力和每天放任，一年下来的能力值相差多少呢?

```
dayup = (1.0 + 0.001)**365              # 提高0.001
daydown = (1.0 - 0.001)**365            # 放任0.001
print("向上: {:.2f}, 向下: {:.2f}.".format(dayup, daydown))
```

运行结果如下：向上：1.44，向下：0.70

每天努力1‰，一年下来将提高44%，好像不多？

继续分析……

例5.2 天天向上 v2.0

一年365天，如果好好学习时能力值相比前一天提高5‰，当放任时相比前一天下降5‰。

效果相差多少呢？

```
dayup = (1.0 + 0.005)**365              # 提高0.005
daydown = (1.0 - 0.005)**365            # 放任0.005
print("向上: {:.2f}, 向下: {:.2f}.".format(dayup, daydown))
```

运行结果如下：向上：6.17，向下：0.16

每天努力5‰，一年下来将提高6倍！这个，不容小觑了吧?

例5.3 天天向上 v3.0

一年365天，如果好好学习时能力值相比前一天提高1%，当放任时相比前一天下降1%。效果相差多少呢？

```
dayfactor = 0.01
dayup = pow((1.0 + dayfactor), 365)          # 提高dayfactor
daydown = pow((1.0 - dayfactor), 365)        # 放任dayfactor
print("向上: {:.2f}, 向下: {:.2f}.".format(dayup, daydown))
```

运行结果如下：向上：37.78，向下：0.03

每天努力1%，一年下来将提高37倍。这个相当惊人吧!

例5.4 天天向上 v4.0

一年365天，一周5个工作日，如果每个工作日都很努力，可以提高1%，仅在周末放任一下，能力值每天下降1%，效果如何呢?

```
dayup, dayfactor = 1.0, 0.01
for i in range(365):
    if i % 7 in [6, 0]:     #周六周日
        dayup = dayup * (1 - dayfactor)
    else:
        dayup = dayup * (1 + dayfactor)
print("向上5天向下2天的力量: {:.2f}.".format(dayup))
```

每周努力5天，而不是每天，一年下来，水平仅是初始的4. 63倍！与每天坚持所提高的37倍相去甚远

例5.5 天天向上 v5.0

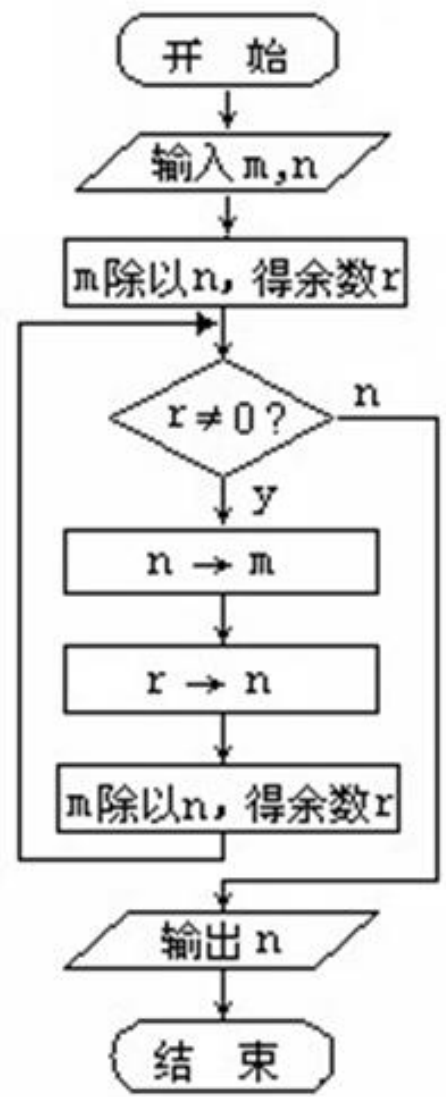
如果对例5.4的结果感到意外，那自然会产生如下问题：每周工作5天，休息2天，休息日水平下降0.01，工作日要努力到什么程度一年后的水平才与每天努力1%所取得的效果一样呢?

```
dayup, dayfactor  = 1.0,  0.001
while(dayup<37.78):
    dayup = 1.0
    for i in range(365):
        if i % 7 in [6, 0]: #周六周日
            dayup = dayup * (1 - 0.01)
        else:
            dayup = dayup * (1 + dayfactor)
    dayfactor += 0.001
print("每天的努力参数是: {:.3f}.".format(dayfactor))
```

如果每周连续努力5天，休息2天，为了达到每天努力1%所达到的的水平，则就需要在工作日将提高的程度达到约2%，即要努力1倍才仅是为了休息2天。

例5.7 最大公约数和最小公倍数

提示：在循环中，只要除数不等于0，用较大数除以较小的数，将小的一个数作为下一轮循环的大数，取得的余数作为下一轮循环的较小的数，如此循环直到较小的数的值为0，返回较大的数，此数即为最大公约数，最小公倍数为两数之积除以最大公约数



```
m = int(input("输入第一个数:  "))
n = int(input("输入第二个数:  "))
t = m*n
if m<n :
   m,n = n,m
while m%n != 0 :
    r = m%n
    m = n
    n = r
print("最大公约数: ",n)
print("最小公倍数:",t//n)
```

当输入的两正整数分别是：432、3584，程序的运行结果是：
最大公约数: 16
最小公倍数: 96768

例5.8：求Fibonacci数列的前20项，并输出。

Fibonacci数列：0,1,1,2,3,5,8,13,21……

f(0)=0,f(1)=1,f(n)=f(n-1)+f(n-2)(n≥2)

分析：数列的前20项包括第0项到第19项。

输出结果：

0	1	1	2	3
5	8	13	21	34
55	89	144	33	377
610	987	1597	2584	4181

```
a, b = 0, 1
for i in range(20) :
    if (i+1)%5!=0 :
        print(a, end='\t')
    else :
        print(a, end='\n')
    a, b = b, a+b
print()
```

例5.9 百鸡百钱

用“枚举法”求解百元买百鸡问题。假定公鸡5元1只，母鸡3元1只，小鸡1元3只，现在有100元钱要买100只鸡，且需包含公鸡、母鸡和小鸡，编程列出所有可能的购鸡方案。

分析：“枚举法”也称为“穷举法”，即将可能出现的各种情况一一进行测试，判断是否满足条件。设公鸡、母鸡、小鸡各有x、y、z只，根据题目要求，可列出方程：

$$\begin{cases} x + y + z = 100 \\ 5x + 3y + z/3 = 100 \end{cases}$$

程序的运行结果如下：

公鸡数	母鸡数	小鸡数
4	18	78
8	11	81
12	4	84

```
print("公鸡数","母鸡数","小鸡数", sep='\t')
for x in range(1,20) :
    for y in range(1,32) :
        z = 100-x-y
        if 5*x+3*y+z/3==100 :
            print(x,y,z, sep='\t')
```