

实验2 程序控制结构的设计

1. 实验目的

熟悉程序的三种基本结构，利用基本的程序结构（顺序、选择、循环）进行小规模程序编写，解决一些小的计算问题。

2. 实验任务

实验任务 2-1 刘同学家的电费

题目描述：

天气转暖，各家各户的用电量都增加了许多。刘同学家今天收到了一份电费通知单，上面写：月用电量在 150 千瓦时及以下部分按每千瓦时 0.4463 元执行，月用电量在 151~400 千瓦时部分按每千瓦时 0.4663 元执行，月用电量在 401 千瓦时及以上部分按每千瓦时 0.5663 元执行。刘同学请你帮忙，编写一个程序，验证一下电费单上的应缴电费是否正确。请编写一个 python 程序，根据已有说明，计算出应缴纳的电费是多少。

输入格式：

输入一个正整数，表示用电总量(单位以千瓦时计)，不超过 10000。

输出格式：

输出一个小数，保留到小数点后 3 位(单位以元计)。

实验指导：

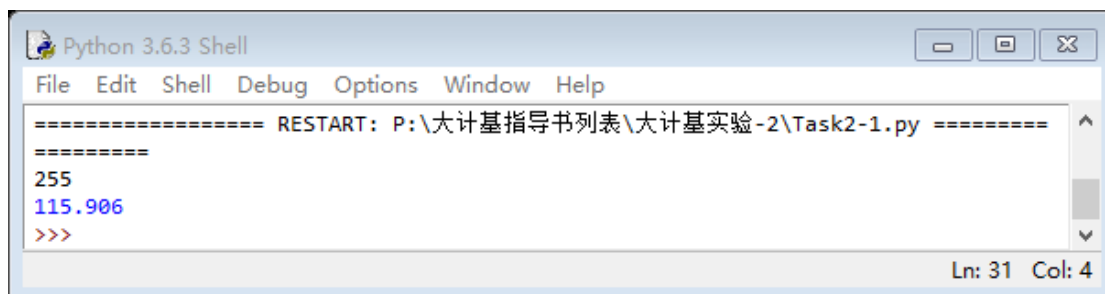
本题考查程序控制结构的选择结构。python 中以缩进区分结构内的语句。python 中条件结构语句为 if-else 结构。如果条件成立，则执行 if 内的语句；如果不成立，则执行后面的 else 或者 elif，else 的后面的语句表示不满足 if 条件而执行的内容，而 elif 相当于 else 和 if 的合并，即不满足 if 的条件后再做判断，若满足条件则执行内部语句。如果 if 后没有 elif 或者 else，则不作任何操作，继续执行整个语句结构之后的内容。if 结构内的语句可以有多行，以缩进作为标志。例如：

```
x = int(input())
if x > 1:
    x = x + 1
    print(x)
elif x < 1:
    x = x - 1
    print(x)
else:
    print(x)
```

上述程序中，如果 x 大于 1，则 x 自加 1，再输出 x ；否则若 x 小于 1，则 x 自减 1，再输出 x ；其他情况下（在这里相当于 x 等于 1），将不进行自加 1 的操作而直接输出。

在本题中，同学们需要分类考虑输入的值的可能范围，再输出合适的答案。

参考运行结果：



实验任务 2-2 冰雹猜想

题目描述：

70 年代中期，美国各所名牌大学校园内风靡一种数学游戏，任意写出一个正整数 N ，并且按照以下的规律进行变换：

如果是个奇数，则下一步变成 $3N+1$ ；

如果是个偶数，则下一步变成 $N/2$ 。

人们发现，无论 N 是怎样一个数字，最终都无法逃脱回到谷底 1。

请编写程序，输入正整数 N ，依据以上规则，输出每步计算的结果（整型变量），到 1 为止。

输入格式：

输入数据包含一行，包含一个正整数，为 N 的值。

输出格式：

输出数据包含若干行，每行包含一个整数，为每步运算的结果，直到结果为 1。

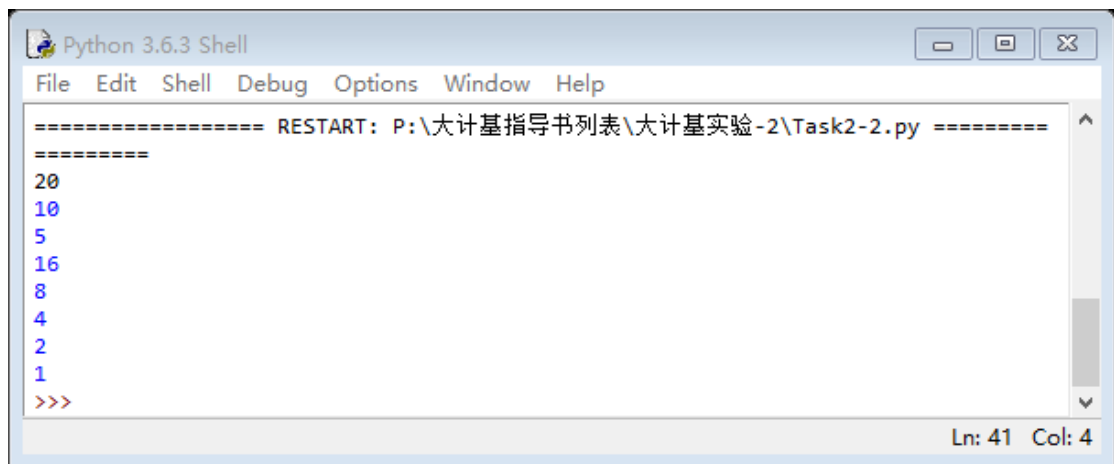
实验指导：

1. Python 中 % 符号代表取余，可用此判断奇偶性。
2. 本题考查程序控制结构的循环结构。Python 中的一种循环结构语句为 while。下面代码段是 while 循环的一种用法：

```
x = 1  
  
while(x < 10000):  
    x = x * 2  
  
print(x)
```

上述程序可以实现计算大于等于 10000 的最小的 2 的幂次方的功能。首次进入循环时，x 会被用于与 10000 比较，由于 x 小于 10000，进入循环。在循环内 x 乘 2 并赋值给自身，因此一次循环后，x 值为 2。之后再次与 10000 比较，依然满足条件，继续循环，直到 x 达到 16384，此时不再满足循环继续条件，退出循环，并输出结果 16384。

参考运行结果：



```
Python 3.6.3 Shell  
File Edit Shell Debug Options Window Help  
===== RESTART: P:\大计基指导书列表\大计基实验-2\Task2-2.py =====  
=====  
20  
10  
5  
16  
8  
4  
2  
1  
>>>
```

实验任务 2-3 TD 线

题目描述:

快期末了, 小天同学的 48 次 TD 线还没刷满, 急的他满头大汗。现在已经过去了 N 周 ($1 \leq N \leq 15$), 前 N 周他每周分别刷了 $x_1, x_2 \dots x_N$ 次 TD, TD 线第 15 周结束时就要关闭了。已知每周最多刷 10 次 TD, 请你帮助小天计算能否按时完成任务。如果能完成, 请输出每周至少需要刷的 TD 次数 (为一个整数), 如果不能按时完成, 请输出 'gg'。

输入格式:

输入数据包含 $N+1$ 行。

第一行为一个正整数 N 。

第 2 至 $N+1$ 行为 N 个正整数 $x_1, x_2 \dots x_N$ 。

输出格式:

输出数据包含一行, 如果能完成, 输出每周至少需要刷 TD 线的次数 (为一个整数); 如果不能完成, 输出 'gg' (小写字母无其他字符)。

实验指导:

1. Python 中的另一种循环结构语句为 `for`。python 中可以使用 `range` 表示从一个整数到另一个整数一个范围 (从起始点开始, 并小于终止点)。`range` 函数经常与 `for` 连用, 用以在该范围内循环迭代, 通常使用方法为“`for i in range(begin, end, step):`”, 其中 `i` 为用于遍历的变量, `begin` 为起始点, `end` 为终止点, `step` 为步长, 即每次遍历的差值 (不能为 0)。若省略 `begin`, 则默认下界为 0; 若省略 `step`, 则默认差值为 1。当起始点 `begin` 大于终止点 `end` 且步长 `step` 不为负, 或是起始点 `begin` 小于终止点 `end` 且步长 `step` 为负时, 该语句不遍历任何内容直接退出。例如:

```
for i in range(1, 6):
```

```
    print(i)
```

表示使 `i` 依次取 1,2,3,4,5;

```
for i in range(4):
```

```
    print(i)
```

表示使 `i` 依次取 0,1,2,3;

```
for i in range(1, 6, 2):
```

```
    print(i)
```

表示使 *i* 依次取 1,3,5:

```
for i in range(6, 1):  
    print(i)
```

或

```
for i in range(1, 6, -2):  
    print(i)
```

将不执行语句块中任何语句，直接向下执行。下面代码段是 **for** 循环的一种用法：

```
total = 0  
for i in range (1, 101):  
    total = total + i  
print(total)
```

上述程序可以实现计算 $1+2+3+\dots+100$ 的功能。*i* 为循环变量，首次进入循环时，*i* 会被赋值为 1，也就是循环起始点，然后依次执行循环内的语句，也就是将 *i* 的值 1 累加到 **total** 变量中，此时第一次循环结束。然后执行下一次循环，因为没有设定步长，步长默认为 1，*i* 的值会自加 1，此时 *i* 为 2，**total** 值再加 2，第二次循环结束时，**total** 的值为 3。直到 *i*=100 累加后，**range** 生成的序列遍历完毕，获得最终结果 5050。注意循环区间为前闭后开，即循环成立条件为： $1 \leq i < 101$

2. 在判断两数是否相等时，应使用“==”而非“=”。例如，以下代码表达当 **a** 等于 3 时，输出 **a**：

```
if a == 3:  
    print(a)
```

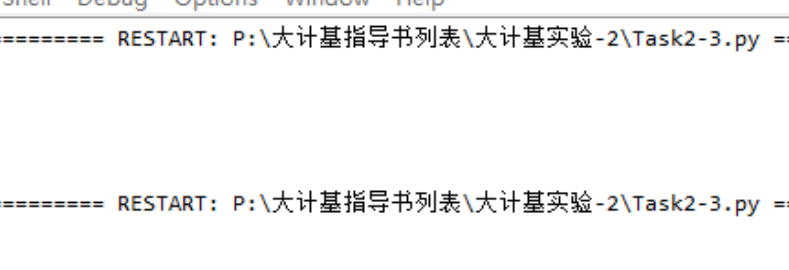
3. **break** 语句和 **continue** 语句可以在 **for** 循环结构与 **while** 循环结构中使用：

当程序运行到 **break** 语句时，将退出所在的该层循环，继续执行循环后的语句；当程序运行到 **continue** 的时候，程序将结束本次循环，回到循环开头继续准备执行下一次循环（若此时是 **for** 循环的最后一次循环或已不满足 **while** 循环条件则退出循环）。**break** 和 **continue** 语句一般包含在循环中的条件判断中，用于在满足某种条件的时候退出循环或结束本次循环。

循环结构也有 **else** 语句，在当循环正常结束的时候执行。如果循环通过 **break** 语句结束，其 **else** 语句就不会执行。下面的程序提供了一种计算 2 到 10 中所有质数的方法：

```
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            #找到 n 的一个因数
            break
    else:
        #遍历完毕未找到 n 的因数完成循环，即 n 为质数
        print(n)
```

参考运行结果:



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
===== RESTART: P:\大计基指导书列表\大计基实验-2\Task2-3.py =====
=====
1
3
4
>>>
===== RESTART: P:\大计基指导书列表\大计基实验-2\Task2-3.py =====
=====
12
0
0
0
0
0
0
2
2
2
2
2
2
88
>>>
Ln: 62 Col: 4
```

实验任务 2-4 输出菱形

题目描述:

编写程序，实现输出一个由空格和星号*组成的菱形，如样例所示。

对于一个 K 行的菱形，第一行第 $\lfloor \frac{K+1}{2} \rfloor$ 个字符为*，前面为空格。第二行从第 $\lfloor \frac{K-1}{2} \rfloor$ 个字符开始的 3 个字符为*，此前为空格。以此类推直到第 $\lfloor \frac{K+1}{2} \rfloor$ 行为 K 个*后按规律减少。

输入格式:

输入数据包含一行，为菱形总共的行数 K ，保证 K 为奇数。

输出格式:

输出数据包含 K 行，为题目要求的菱形。

实验指导:

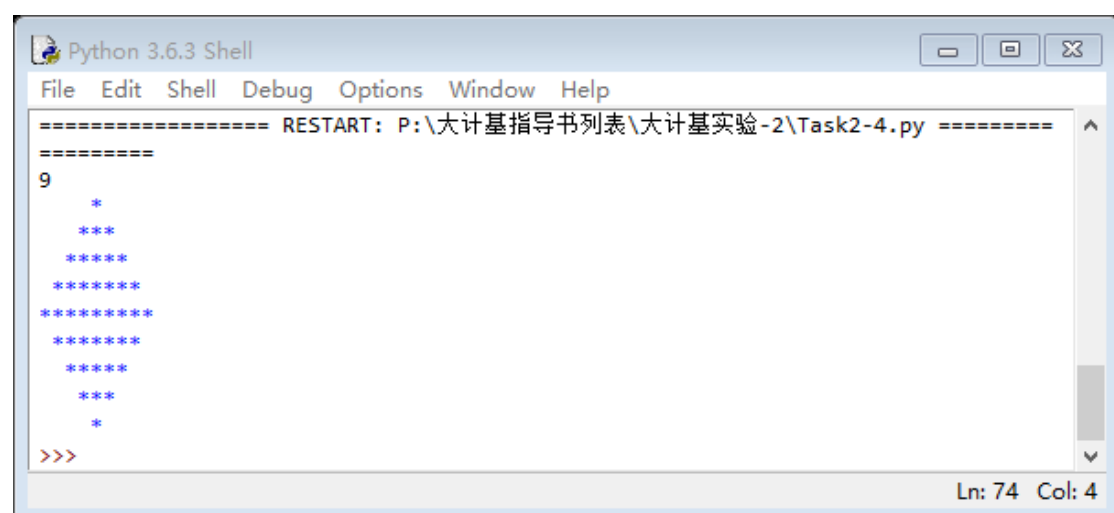
Python 中每次 `print()` 输出默认换行，如果想在输出后不换行，可以通过添加一个 `end=""` 参数使结束符为空字符串以便使 `print()` 不换行。如：

```
for i in range(5):  
    print(i, end="")
```

该代码会不换行地输出连续的 12345。

如果想要在已有 `end=""` 后的下一个 `print()` 换行，使用 `print("\n")`，输出一个空字符串，通过默认存在的换行即可完成换行。

参考运行结果:



```
Python 3.6.3 Shell  
File Edit Shell Debug Options Window Help  
===== RESTART: P:\大计基指导书列表\大计基实验-2\Task2-4.py =====  
=====  
9  
    *  
   ***  
  *****  
 *****  
*****  
*****  
 *****  
  *****  
   ***  
    *  
>>>
```

Ln: 74 Col: 4

实验任务 2-5 质因数分解（选做）

题目描述：

每个合数都可以写成几个质数相乘的形式，其中每个质数都是这个合数的因数，叫做这个合数的分解质因数。如 504 的质因数分解为 $504 = 2^3 \times 3^2 \times 7$

请你编写程序，从键盘读取一个大于 1 的正整数，将其分解质因数输出。

输入格式：

输入一个正整数 x 。保证 x 为不大于 1000 的合数。

输出格式：

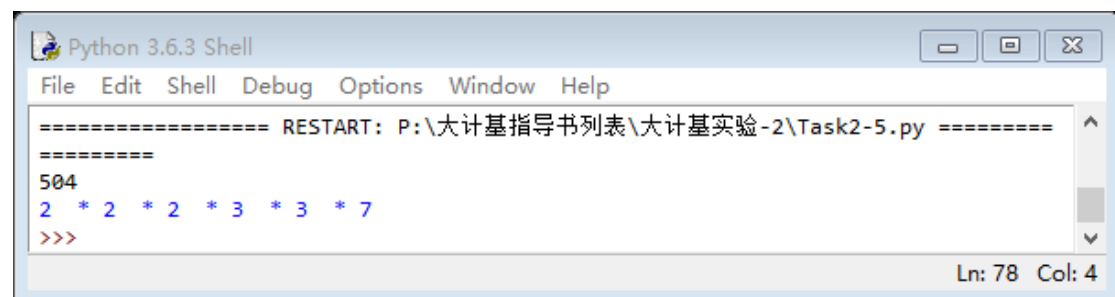
输出数据包含一行，包含各个质因数，各个质因数之间用“*”连接，在每个数字和每个“*”之间都包含一个空格。

实验指导：

请思考：

1. 如何发现所要求数的因数？应当以怎样的顺序去寻找？
2. 是否需要判断一个因数是否为质数？如果是，如何判断？如果不是，为什么？
3. 如何按要求输出正确结果？

参考运行结果：



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
===== RESTART: P:\大计基指导书列表\大计基实验-2\Task2-5.py =====
=====
504
2 * 2 * 2 * 3 * 3 * 7
>>>
Ln: 78 Col: 4
```