

模拟一

用Python编辑器打开“99”下的文件“水仙花数.py”，进行以下操作并保存结果。

- (1) 请填写完善该程序，实现功能输出所有水仙花数。水仙花数是指一个三位数它的每一位上的数字的3次方之和等于它本身。例如：三位数153满足： $1^3+5^3+3^3=1+125+27=153$ ，则153是水仙花数。设变量i, j, k分别用于存放个位、十位和百位上的数。
- (2) 编写完成后原名保存并关闭应用软件。

#请删除序号及下划线，填写正确的代码，使程序完善。

```
for x in range(100,1000):
    i=___①___           #个位数
    j=(x//10)%10       #十位数
    k=___②___           #百位数
    if x==___③___:
        print("水仙花数是",x)
```

用Python编辑器打开“124”下的文件“背诵古诗.py”，进行以下操作并保存结果。

- (1) 中国古诗词是中华民族的优秀文化遗产，蕴含丰富的人民智慧，背诵古诗词好处很多，诗词社团的同学编写了一个背诵古诗的程序，首先利用变量，在程序中存入完整的古诗内容，要求用户循环背诵古诗，指导背诵的版本和正确版本完全一致，退出循环。
- (2) 当用户背诵次数超过3，对其进行鼓励加油。
- (3) 编写完成后原名保存并关闭应用软件。

```
#背诵古诗
#存储古诗内容
poem = '空山新雨后，天气晚来秋。明月松间照，清泉石上流。'
yourpoem = input("请背诵《山居秋暝》前两句内容\n")
count = ___①___ # 设置 count 变量用来统计背诵次数
while(poem ___②___ yourpoem):
    print('你还没有完成背诵，请再试一次！')
    if count > ___③___:
        print('继续加油！相信自己！')
    yourpoem = input()
    count = ___④___
print("背诵正确！恭喜成功！")
```

模拟二

用Python编辑器打开“103”下的文件“计算1-2+3-4...+99-100的值.py”，进行以下操作并保存结果。

- (1) 请填写完善该程序，实现功能：计算表达式  $s=1-2+3-4\cdots+99-100$  的值并输出结果；
- (2) 编写完成后原名保存并关闭应用软件。

```
s=0
for i in range(1,___①___):
    if ___②___:
        s=s-i
    else:
        ___③___
print(___④___)
```

用Python编辑器打开“120”下的文件“求pi的近似值.py”，进行以下操作并保存结果。

- (1) 填空完善该程序，实现功能：尝试求解  $\pi$  的近似值（求解关系式为： $\frac{\pi}{4}=1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots$ ）。
- (2) 编写完成后原名保存并关闭应用软件。

```
sum,a,b,t=0.0,1,1.0,1.0 #给各参数依次赋值
while b<1000:
    sum=sum+___①___
    b=___②___
    a=___③___
    t=a/b
pi = ___④___
print("pi 的值是{:.20f}".format(pi)) # 输出 20 位浮点型数值
```

### 模拟三

用Python编辑器打开“104”下的文件“找出三个数中最大的数.py”，进行以下操作并保存结果。

(1) 请填空完善该程序，实现功能：输入任意三个不同的整数，输出其中最大的一个数；

(2) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
a=int(input("请输入第一个数："))
__①__=int(input("请输入第二个数："))
c=int(input("请输入第三个数："))
if a>b:
    __②__
else:
    max=b
if __③__:
    max=c
print(__④__)
```

用 Python 编辑器打开“122”下的文件“输出 1 千到 1 万之间的素数.py”，进行以下操作并保存结果。(1)素数也叫质数，是除了 1 和它自身没有其它因数的自然数。

(2)根据数学原理，判断一个数 n 是否为素数只要判断 2 到 n 的算术平方根之间是否存在 n 的因数，该算法对于大整数的判断速度有明显提高。请填空完善该程序，实现功能：求 1000 到 10000 之间的素数。(3)编写完成后原名保存并关闭应用软件。

```
def isPrime(n): #定义函数 isPime 判断参数 n 是否为素数，是返回 True，否则返回 False
    if n <= 1: #小于等于 1 的数不是素数
        return False
    for i in range(2, int(__①__)+ 1): # i 的范围 2 到 n 的平方根取整，平方根用 n
        的 0.5 次方计算
        if __②__: # 如果 i 是 n 的因数，则返回 False
            return False
    return True
for x in range(1000,10001):
    if __③__: # 调用函数进行判断 x 是否为素数
        print(x)
```

### 模拟四

用Python编辑器打开“100”下的文件“求阶乘和.py”，进行以下操作并保存结果。

(1) 请填空完善该程序，实现功能输出： $s=1!+2!+3!+4!+5!$  说明： $n!=1\times 2\times 3\times 4\times \dots\times n$ ，例如： $3!=1\times 2\times 3$ 。

(2) 编写完成后原名保存并关闭应用软件。

```
s=0
n=__①__
for i in range(1,__②__):
    n=n*i
    s=__③__
print("1!+2!+3!+4!+5!="s)
```

用Python编辑器打开“123”下的文件“能力值.py”，进行以下操作并保存结果。

(1) 俗话说“日有所进，月有所长”，假设每个人的初始能力值是1，经过一天的努力学习和工作后，能力会比之前增长1%。

(2) 如果周一到周五努力学习提升能力，而周末休息，且休息一天能力就降低1%。同时通过取余运算来判断某一天是学习日还是休息日，如果余数是0或者6，那么这一天就是休息日；余数是其他数值时，这一天是学习日。

(3) 请完善程序，计算两种情况下，一年后能力值增长情况。

```
#天天向上
dayup=1 #设置 dayup 为能力值， factor 为能力变化幅度
factor=0.01
for i in range(0,__①__):
    dayup = __②__ * (1+factor)
print("天天向上的能力值： %.2f" % dayup)

#五上二下
dayup=1
for i in range(__③__):
    if __④__ in [6, 0]: # 判断是否为休息日，是则能力下降；否则能力上升。
        dayup = dayup*(1-factor)
    else:
        dayup = dayup*(__⑤__)
print("向上 5 天向下 2 天的能力值： %.2f" % dayup) # 打印结果
```

<p><b>模拟五</b></p>	
<p>用 Python 编辑器打开“106”下的文件“统计一串字符中数字的个数.py”，进行以下操作并保存结果。(1)请填写完善该程序，实现功能:输入一串字符，用回车结束，统计其中数字出现的次数:(2)编写完成后原名保存并关闭应用软件。</p>	<pre>a=str(input("请输入一串字符,用回车结束: ")) t=__①__ for i in a:     if(i&gt;='0' __②__ i&lt;='9'):         t=__③__ print(__④__)</pre>
<p>用 Python 编辑器打开“119”下的文件“求最小公倍数.py”，进行以下操作并保存结果。 (1)算法描述:最小公倍数初始值设为较大数，如果能整除，则较大数就是最小公倍数;如果不能整除，则让较大数乘以2,3,4...递增1的自然数，直到能整除为止。如:除入“11,5”时，输出结果为“55”，输入“8，12”时，输出结果为“24”； (2)请填写完善该程序，实现功能:键盘上输入两个自然数 m 和 n(1&lt;n&lt;=m&lt;=1000)，输出这两个自然数的最公倍数。</p>	<pre>m=int(input("请输入自然数 m:")) n=int(input("请输入自然数 n:")) i=1 if m&lt;n:     m,n=__①__ #两数交换（用大数翻倍） s=m #s 的初始值较大数 while __②__:     i = i+1     s = __③__ print(__④__)</pre>
<p><b>模拟六</b></p>	
<p>用Python编辑器打开“101”下的文件“反转三位数.py”，进行以下操作并保存结果。 (1) 请填写完善该程序，实现功能：输入一个任意三位整数，将它反转输出，例如369反转成963。 (2) 编写完成后原名保存并关闭应用软件。 注：程序修改时，请把下划线及序号删除；不能删除注释语句。</p>	<pre>x=__①__(input("请输入一个任意三位整数: ")) ge=__②__ shi=x//10%10 bai=__③__ __④__=ge*100+shi*10+bai print(y)</pre>
<p>用 Python 编辑器打开“121”下的文件“尼拉坎特哈级数求兀近似值.py”，进行以下操作并保存结果。 (1)计算兀的方法有很多种，印度数学家尼拉坎特哈发现了一个可用于计算兀的无穷级数，其展开公式如下： <math display="block">\pi = 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \frac{4}{12 \times 13 \times 14} \dots</math> (2)请观察各分数项符号和分母的规律，编写代码实现求项数为 100 时的兀近似值。(3)编写完成后原名保存并关闭应用软件。</p>	<pre>op=1 #op 保存分数项前的符号，取值为 1 或-1 pi=3 for i in range(2, 101, __①__):     pi = pi + __②__ # 循环迭代求 pi 的近似值     op = - __③__ # 分数项前的符号系数变成原来的相反数 print(__④__)</pre>
<p><b>模拟七</b></p>	
<p>用Python编辑器打开“436”下的文件“偶数和.py”，进行以下操作并保存结果。 (1) 请填写完善该程序，实现功能：“求0~50所有偶数的和”； (2) 编写完成后保存文件并关闭应用软件。 注：程序修改时，请把下划线及序号删除；不能删除注释语句。</p>	<pre>#求 0~50 所有偶数的和 s = __①__ for i in range(0, __②__, __③__):     s = s + __④__ print("0~50 所有偶数的和为", __⑤__)</pre>

用Python编辑器打开“441”下的文件“冰雹猜想.py”，进行以下操作并保存结果。  
 (1) 冰雹猜想是指：一个正整数，如果是奇数就乘以3再加1，如果是偶数就对它除以2(如下图)，这样经过若干个次数，最终回到1。无论这个过程数值如何庞大，就像瀑布一样坠落，在经过若干次的变换之后也必然会到纯偶数：4-2-1。

$N$ 变为  $\begin{cases} 3N+1 & (\text{如果}N\text{是奇数}) \\ N/2 & (\text{如果}N\text{是偶数}) \end{cases}$

(2) 请填写完善该程序，实现功能：键盘上输入任意一个正整数，将其按冰雹猜想的规则进行拆解，输出每一步拆解的值和总步数，验证其是否符合冰雹猜想。  
 (3) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
x=___①___(input('请输入任一正整数: '))
n=0
while(___②___):
    if(x%2!=0):
        x=___③___
    else:
        x=int(x/2)
    print(___④___,end=' ')
    n=___⑤___
print()
print('总步数: ',n)
```

模拟八

用Python编辑器打开“430”下的文件“工资表.py”，进行以下操作并保存结果。

(1) 公司规定，职员工龄津贴10年以内(含10年)的每月100元，第11至20年每月200元，第21至30年每月300元，工作满30年后每月400元。

(2) 计算职员工作若干年来工龄津贴总和。

(3) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
def gljt(n):
    if ___①___ #10年以内
        s=100*12*n
    elif n<=20: #第11至20年
        s=12000+200*12*(n-10)
    elif n<=30: #第21至30年
        s=36000+300*12*(n-20)
    ___②___ #其他工作年限
        s=72000+400*12*(n-30)
    return ___③___
year=___④___(input("请输入工作年限: "))
print("工龄津贴总和: ",___⑤___)
```

用Python编辑器打开“446”下的文件“算数平方根.py”，进行以下操作并保存结果。

(1) 利用牛顿迭代法，编程计算正数G的算术平方根的近似值(精确到小数点后5位)。  
 牛顿迭代法计算正数G的算术平方根(近似值)的算法如下：

$$x_{i+1} = \frac{1}{2} \left( x_i + \frac{G}{x_i} \right) \quad i = 0, 1, 2, 3 \dots$$

(2) 近似值的初始值  $x_0$  取2。不断迭代更新  $x_i$  的值，直到  $|x_i - x_{i-1}| < 10^{-6}$ ，此时的  $x_i$  就是我们要求的近似值。

(3) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
n = int(input("请输入一个正数: "))
x1=2 #x1 赋初始值 2
x2=1/2*(x1+n/x1) #计算 x2 的值
if x1==x2:
    print(n,"的算数平方根是 %.5f"%x2) #输出 n 的算数平方根，保留 5 位小数
else:
    while abs(x2-x1)>=1e-6: #当|x1-x2|大于等于 10 的-6 次方时，进行迭代
        x1 = ___①___ # 将 x2 的值赋给 x1
        x2 = ___②___ # 重新计算 x2 的值
    print(n,"的算数平方根是 %.5f"___③___) # 输出 x2 的值，保留 5 位小数
```

模拟九

<p>用Python编辑器打开“433”下的文件“进制转换.py”，进行以下操作并保存结果。</p> <p>(1) 通常我们可以采用“除2取余，倒序输出”的方法，将十进制整数转换为二进制数，例如11除以2，商是5，余数为1；再用5除以2，商是2，余数为1……，依次类推，最后将余数倒序输出即为二进制数1011。</p>	<pre>#十进制转二进制 n=__①__(input('请输入一个大于0的十进制整数: ')) s="" #将字符串s赋值为空 while n&gt;0:     if n%2==0: #如果n除以2的余数为零         __②__ #把字符'0'添加到字符串s的最前面     else:         s='1'+s #把字符'1'添加到字符串s的最前面         __③__ #将n整除以2,并重新赋值给n print('转换之后的二进制数: %s'%s)</pre>
<p>用Python编辑器打开“450”下的文件“四位数.py”，进行以下操作并保存结果。</p> <p>(1) 某个四位数有如下特点：①这个数加1之后是15的倍数；②这个数减去3是38的倍数；③把这个数各数位上的数左右倒过来所得的数与原数之和能被10整除，求这个四位数。</p>	<pre>def dx(num): #数字倒序函数     num1 = __①__ #num1赋初值     while num != 0:         gw = __②__ #得到个位上的数         num1 = num1*10+gw         num=num//10     return __③__ #返回倒序后的数值 '''判断四位数是否满足下列条件，满足则输出： 加1为15的倍数，减3为38的倍数， 与本身倒序的四位数相加的和则为10的倍数''' for i in range(1000, 10000):     if (i+1) % 15 == 0 and __④__ and (i+dx(i)) % 10 == 0:         print(i)</pre>
<p>模拟十</p>	
<p>用Python编辑器打开“428”下的文件“计算三角形内切圆面积.py”，进行以下操作并保存结果。</p> <p>(1) 三角形内切圆半径<math>r=s/p</math>，<math>s</math>为三角形面积，<math>p</math>为三角形周长的一半。</p> <p>(2) 海伦公式求三角形面积</p> $S = \sqrt{p * (p - a) * (p - b) * (p - c)}$ ， $a, b, c$ 为	<pre>from math import * while True:     a=__①__(input("请输入a的边长: "))     b=float(input("请输入b的边长: "))     c=float(input("请输入c的边长: "))     #判断边长是否构成三角形     if a &lt;= 0 or b &lt;= 0 or c &lt;= 0 or(a+b &lt;= c) or (a+c &lt;= b) or (__②__):         __③__("边长输入有误,请重新输入。")     __④__:         break     p=(a+b+c)/2     s=sqrt(p*(p-a)*(p-b)*(p-c))     r=s/p     cs=pi*r*r     print("三角形内切圆的面积: %.2f" % __⑤__) #打印三角形内切圆面积,保留2位小数</pre>

用Python编辑器打开“440”下的文件“打印图形.py”，进行以下操作并保存结果。

(1) 请填空完善该程序，实现功能：打印由n行短横线 and 星号组成的等腰三角形，n由键盘输入。

(2) 测试程序：输入数字5，则打印出如下图形。

输入三角形行数：5

```
-----*
---***---
--*****--
-*****-
*****
*****
```

```
n = ___①___(input('输入三角形行数: ')) #三角形的行数

for i in range(n):
    print('-'*(n-i-1)+'*'*( ___②___)+ ___③___)
```

### 模拟十一

用Python编辑器打开“443”下的文件“素数.py”，进行以下操作并保存结果。

(1) 素数又叫质数，是指除了1与本身以外没有其他因数的数，2是自然数中最小的素数。

(2) 请填空完善该程序，实现功能：键盘输入一个自然数n(n<1000)，输出1至n范围内的所有素数。

```
n=int(input("输入自然数 n="))
for i in range(2,___①___): #枚举范围内的每一个数字
    flag=0
    for j in range(2,i):
        if ___②___: #判断 i 是否为素数
            flag=1
            ___③___ #退出循环
    if flag==0:
        print(i,end=' ')
```

用 Python 编辑器打开“439”下的文件“因子个数.py”，进行以下操作并保存结果。(1)编写程序实现以下功能;输入一个正整数，统计该数因子的个数。(2)编写完成后原名保存并关闭应用软件。

```
x=int(input("请输入一个正整数: "))
s=___①___ #存放因子的变量初始化
for i in range(1,___②___):
    if(___③___): #判断是否是因子
        s=___④___
print(___⑤___)
```

### 模拟十二

用 Python 编辑器打开“431”下的文件“降雨分类.py”，进行以下操作并保存结果。

(1)气象学上，降水等级的划分一般是根据 24 小时内的降水量来确定，具体数量关系如下表所示。

(2)已知某市各个地区的 24 小时降水量，存放在列表 rainfall 中，请完善程序，使其能够根据不同地区的降雨量数值，输出对应气象学分类。

(3)编写完成后原名保存并关闭应用软件。

注:程序修改时，请把下划线及序号删除;不能删除注释语句。

24小时降水量	气象学分类
小于10毫米	小雨
大于等于10毫米，小于25毫米	中雨
大于等于25毫米，小于50毫米	大雨
大于等于50毫米，小于100毫米	暴雨
大于等于100毫米，小于250毫米	大暴雨
大于等于250毫米	特大暴雨

```
#根据各地区的降雨量输出降雨分类
rainfall=[32.1,6.5,66.8,16.5,266.6] #雨量列表
for i in ___①___: #对雨量列表进行循环判断
    if i<10:
        print('小雨')
    elif i<25:
        print('中雨')
    elif i<50:
        print('大雨')
    elif i<100:
        print('暴雨')
    ___②___:
        print('大暴雨')
    else:
        print(___③___)
```

<p>用Python编辑器打开“444”下的文件“年龄.py”，进行以下操作并保存结果。</p> <p>(1)有5个人坐在一起，问第5人多少岁？他说比第4个人大两岁。问第4人的岁数？他说比第3个人大两岁。问第3个人，又说比第2个人大两岁。问第2个人，他说比第1人大两岁。最后问第1个人，他说10岁。请问第五个人多大？</p>	<pre>def age(n):          #递归求年龄     if __①__:         return 10     else:         return __②__ #前面一个人的年龄加 2  print("第 5 个人的年龄是：",__③__)</pre>
<p><b>模拟十三</b></p>	
<p>用 Python 编辑器打开“442”下的文件“最小公倍数.py”，进行以下操作并保存结果。</p> <p>(1)计算两个正整数的最小公倍数有多种方法，其中有一种方法是用这两个数的乘积除以这两个数的最大公约数，而最大公约数则可以采用辗转相除法来计算。(2)请填空完善该程序，实现功能:两个正整数的最小公倍数。(3)编写完成后原名保存并关闭应用软件。</p>	<pre>#最小公倍数 m=int(__①__('请输入第一个正整数: ')) n=int(input('请输入第二个正整数: ')) mn=m*n          #将 m 与 n 的乘积赋值给变量 mn __②__          #计算 m 除以 n 的余数 while r!=0:    #辗转相除法求最大公约数     m=n     n=r     r=m%n print(__③__) #输出 m、n 的最小公倍数</pre>
<p>用 Python 编辑器打开“438”下的文件“随机数加法.py”，进行以下操作并保存结果。</p> <p>(1)编写程序实现以下功能:随机产生两个 20 以内的加数，请你输入计算结果，计算正确则输出“计其正确!”，结果不正确则输出“计算错误!”。(2)编写完成后原名保存并关闭应用软件。</p>	<pre>import __①__ #导入随机库模块 a=random.randint(1,20) b=random.randint(1,20) print(a,"+",b,"=") __②__=int(input("请你输入计算结果: ")) __③__ (__④__==c):     print("计算正确! ") __⑤__:     print("计算错误! ")</pre>
<p><b>模拟十四</b></p>	
<p>用 Python 编辑器打开“452”下的文件“最大公约数 2.py”，进行以下操作并保存结果。</p> <p>(1)自定义递归函数，利用欧几里得辗转相除法求两个数的最大公约数。辗转相除法是求最大公约数的一种方法，具体做法是:两个数中用较大数除以较小数，再用出现的余数(第一余数)去除除数，再用出现的余数(第二余数)去除第一余数，如此反复，直到最后余数是 0 为止。得到最后的除数就是这两个数的最大公约数。</p> <p>(2)编写完成后原名保存并关闭应用软件。</p>	<pre>def gcd(a,b):      #递归法求最大公约数     r=__①__        #求余数     if r==0:         return b   #余数为零返回最大公约数     else:         return gcd(__②__)  m=int(input("m=")) n=int(input("n=")) print(m,n,"的最大公约数为",gcd(__③__))</pre>
<p>期末考试结束了，请帮任课老师编写一个程序，用 Python 编辑器打开文件“432”下的文件“成绩.py”，进行以下操作并保存结果。(1)请填空完善该程序，实现功能:依次输入人数和每一位同学的成绩，即可输出全班最高分、最低分和平均分。(2)编写完成后原名保存并关闭应用软件。</p>	<pre>n=__①__(input('请输入全班人数: ')) s=0      #总分初值 x=0      #最高分初值 y=999    #最低分初值 for i in range(1,__②__):     print(i,'号得分: ')     a=float(input())     s=__③__</pre>

	<pre>x=max(x,a) y=min(④,a) print('最高分: ',x) print('最低分: ',y) print('平均分: ',⑤)</pre>
<p><b>模拟十五</b></p>	
<p>用 Python 编辑器打开“447”下的文件“素数个数.py”，进行以下操作并保存结果。(1)请填空完善该程序，实现功能:编写程序输出 10000 以内的素数的个数。(2)编写完成后原名保存并关闭应用软件。</p>	<pre>① prime(n): # 定义函数，判断是素数返回 True，否则返回 False     for i in range(2, n//2+1):         if ②: # n 能够整除 i             return False # 返回 False     return True k=0 for i in range(2,10000):     if prime(i): #判断是否是素数         k += ③ # 计数 k 加 1 print("10000 以内的素数的个数是: ", ④)</pre>
<p>用 Python 编辑器打开“435”下的文件“判断.py”，进行以下操作并保存结果。</p> <p>(1)输入两个大于零的整数 a、b，判断 a 与 b 的大小，如果 a 大于 b，则输出 <math>a^2-b</math> 的值，否则输出 a+b 的值。示例：</p> <pre>请输入一个不为零的整数: 6 请输入一个不为零的整数: 9 16 &gt;&gt;&gt;</pre>	<pre>#输入两个不为零的整数 a、b: a=int(input("请输入一个不为零的整数: ")) b=int(input("请输入一个不为零的整数: ")) #在以下区域中按题目要求编写代码实现相应功能。 ① ② ③ ④</pre>
<p><b>模拟十六</b></p>	
<p>李华在某购物平台挑选礼物:围巾 459 元，伴身器 1160 元，保温杯 128 元，羊毛衫 997 元。商家有两种优惠方案，只能任选其一:方案一，满 2 件，总价打 9 折，满 3 件及以上，总价打 8 折;方案二，每 200 减 30(上不封顶)。</p> <p>用 python 编辑器打开“429”下的文件“促销优惠.py”，进行以下操作并保存结果。(1)请填空完善该程序，实现功能:计算两种方案分别应付多少钱。(2)编写完成后原名保存并关闭应用软件。</p>	<pre>things = [459, 1160, 128, 997] #所选商品价格列表 n = len(things) #计算商品数量 total=sum(things) #计算所有商品打折前总价 #方案一 if n&gt;3:     s1=total*① elif n②:     s1=③ else:     s1=total #方案二 s2=total-④*30 print('选择方案一应支付: ',s1,'元') print('选择方案二应支付: ',⑤,'元')</pre>
<p>用 Python 编辑器打开“445”下的文件“进制转换.py”，进行以下操作并保存结果。(1)输入一个十进制正整数，输出该数的十六进制数。例如输入:167，输出:A7。(2)编写完成后原名保存并关闭应用软件。</p>	<pre>n = ① (input("请输入一个十进制正整数: ")) # 十六进制的各个数码表示成列表 num16 = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "A", "B", "C", "D", "E", ②] a = "" while n&gt;0:     r = n % ③ # 对 16 取余数     a = ④ + a # 对应余数，从十六进制的数码列表中取出相应的值</pre>

```
n = n//16
print("该数的十六进制数是: ", __⑤__) # 输出
```

### 模拟十七

用Python编辑器打开“427”下的文件“保护长江濒危动物.py”，进行以下操作并保存结果。

(1) 添加濒危动物“扬子鳄”，将“长尾鲟”修改为“长江鲟鱼”，删除已灭绝的水生动物“长江白鲟”。

(2) 将濒危动物名单打印在同一行。

(3) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
a=["长江白鲟","长尾鲟","长江江豚","中国大鲵","胭脂鱼","长薄鳅","中华鲟","长江鲟"]
a.__①__("扬子鳄")          #添加濒危动物"扬子鳄"
__②__="长江鲟鱼"          #将"长尾鲟"修改为"长江鲟鱼"
__③__ a[__④__]           #删除已灭绝的水生动物"长江白鲟"
for i in a:
    print("{:5}".format(i,__⑤__="")) #按格式打印，不换行
```

用Python编辑器打开“448”下的文件“π的近似值.py”，进行以下操作并保存结果。

(1) 请填空完善该程序，实现功能：根据以下公式求π的近似值。

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + \frac{(-1)^{999999}}{2 \times 999999 + 1}$$

(2) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
pi = 0          #pi 表示圆周率，赋初始值 0
f = 1          #f 用来控制+、-号
for k in range(0, __①__): # range()中是循环的次数，即总项数
    t = 1/(2*k + 1)
    pi = pi + __②__
    f = __③__ # 符号变化，取相反数
print("圆周率 pi = ", __④__) # 输出 pi 的值
```

### 模拟十八

用 Python 编辑器打开“453”下的文件“兔子.py”，进行以下操作并保存结果。

(1) 古典问题:有一对兔子，从出生后第 3 个月每个月都生一对兔子，小兔子长到第三个月后每个三又生一对兔子，假如兔子都不死，问每个月的兔子总数为多少？

(2) 程序分析:兔子的规律为数列:1.1.2.3.5.8.13, 21……此数列称为斐波拉契数列。(3)编写完成后原名保存并关闭应用软件。

```
def fib(n):          #利用迭代求斐波拉契数列的第 n 个数
    r1,r2=1,1        #第 1 个月、第 2 个月初值设定
    for i in range(__①__,n+1):
        r1,r2=r2,__②__
    return r2
month=int(input("输入需要计算的月份数: "))
print("兔子总对数为: ",fib(__③__))
```

用Python编辑器打开“434”下的文件“赋值.py”，进行以下操作并保存结果。

(1) 将数值9赋给变量a；

(2) 将数值3赋给变量b；

(3) 将表达式a/b+a的值赋给变量c；

(4) 将函数float(c)的值赋给变量d；

(5) 使用输出语句在屏幕上输出：d的值为12.0；

(6) 依次编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

```
#按题意写代码
__①__
__②__
__③__
__④__
__⑤__
```

### 模拟十九

用Python编辑器打开“449”下的文件“灯.py”，进行以下操作并保存结果。

(1) 《镜花缘》是清代文人李汝珍创作的长篇小说，书中有这样一个情节：一座楼阁到处挂满了五彩缤纷的大小灯球，灯球有两种，一种是大灯A：下缀2个小灯，另一种是大灯B：下缀4个小灯。

(2) 请填空完善该程序，实现功能：输入大灯和小灯的总数，对输入的数据进行校验，如果输入数据有效，则输出大灯A与大灯B的数量。

(3) 编写完成后原名保存并关闭应用软件。

注：程序修改时，请把下划线及序号删除；不能删除注释语句。

'''《镜花缘》是清代文人李汝珍创作的长篇小说，

书中有这样一个情节:一座楼阁到处挂满了五彩缤纷的大小灯球,  
灯球有两种,一种是大灯 A:下缀 2 个小灯,另一种是大灯 B:下缀 4 个小灯。  
程序实现功能:输入大灯和小灯的总数,对输入的数据进行校验,  
如果输入数据有效,则输出大灯 A 与大灯 B 的数量;数据输入无效,则输出:数据输入有误"

```
num1 = ①__ (input("请输入大灯的数量: ")) # 输入大灯的数量
__②__ = int(input("请输入小灯的数量: ")) # 输入小灯的数量
if num1 > 0 and num2 > num1*2 and num2 < num1*4 and __③__:
    #对数据进行校验: 大灯数量为正数, 并且小灯的数量在大灯的 2 倍与 4 倍之间, 并且为偶数
    numA = (4*num1-num2)//2 # 计算大灯 A 的数量
    numB = __④__ # 计算大灯 B 的数量, 公式为: 大灯的数量-大灯 A 的数量
    print("大灯 A 有: ", numA, "盏, 大灯 B 有", numB, "盏") # 输出大灯 A 和 大灯 B 的数量
else:
    print("数据输入有误") #如果数据有误, 则输出提示语句
```

用Python编辑器打开“437”下的文件“判断闰年.py”,进行以下操作并保存结果。  
(1) 闰年的判断条件是可以被400整除,或者可以被4整除并且不能被100整除。  
(2) 编写自定义函数,请输入一个年份,输出该年份是否是闰年,是则输出“YES”,否则输出“NO”  
(3) 编写完成后原名保存并关闭应用软件。  
注:程序修改时,请把下划线及序号删除;不能删除注释语句。

```
def rn(__①__):
    if(num%400==0 or num%4==0 __②__
num%100!=0):
        return "YES"
    __③__:
        return "NO"
year=__④__(input("请输入年份: "))
print(__⑤__)
```

## 模拟二十

用Python编辑器打开“426”下的文件“车道测速提醒.py”,进行以下操作并保存结果。



(1) 通常情况下,高速公路客车专用道/快车道,限制的机动车辆速度在90-120公里/小时之间。  
(2) 编写自定义函数,实现车道测速提醒功能。  
(3) 编写完成后原名保存并关闭应用软件。  
注:程序修改时,请把下划线及序号删除;不能删除注释语句。

```
__①__ cs(b):
    if b>120:
        print("已超速,请减速!")
    __②__
        print("车速太低,易追尾,请提速!")
    __③__
        print("车速正常,请谨慎驾驶!")
v=__④__(input("请输入车速: "))
cs(__⑤__)
```

用Python编辑器打开“451”下的文件“计算.py”,进行以下操作并保存结果。  
(1) 一个正整数的阶乘是所有小于及等于该数的正整数的积,并且0的阶乘为1,即 $n! = 1 \times 2 \times 3 \times \dots \times (n-1)$

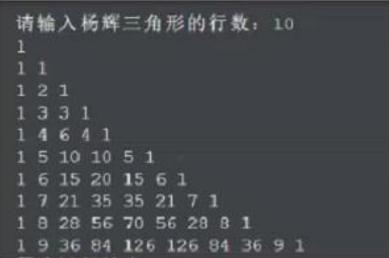
$$n! = \begin{cases} 1 & n=0,1 \\ n*(n-1)! & n>1 \end{cases} \quad A = \frac{3^n}{n!}$$

$\times n$ 。即 现计算  $A = \frac{3^n}{n!}$ , 并输出A的值。

(2) 编写完成后原名保存并关闭应用软件。  
注:程序修改时,请把下划线及序号删除;不能删除注释语句。

```
def jc(n): #利用递归的方法求 n!
    if n == 0 or __①__:
        return __②__
    else:
        return __③__
n=int(input('请输入正整数 n: '))
if n>0: #如果 n 为正数, 且为整数
    A = __④__ #计算 A 的值
    print('A=', A)
else:
    print('输入的数据有误, 无法计算')
```

## 模拟二十一

<p>小明同学想进入某数学学习资源平台，需要输入正确的用户名和密码才能浏览，如果用户名和密码均正确则提示登录成功！，允许浏览，否则显示登录失败，小明同学的登录用户名为xiaoming，密码为123456。用Python编辑器打开“D:\NetStuOp\08\08\652”文件夹下的文件“登录验证.py”，请在序号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre>①= input("请输入您的用户名：") password = input("请输入您的密码：") #用户名是 xiaoming 且密码是 123456 则系统提示登录成功，否则提示登录失败 if admin == "xiaoming" and ②:     print("登录成功!") ③:     print("登录失败!")    #登录失败!</pre>
<p>编程实现输入一个正整数m，使其逆向输出。例如输入正整数：12345，输出：54321。用Python编辑器打开“D:\NetStuOp\08\08\653”文件夹下的文件“正整数逆向输出.py”，请在序号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre>m = ①(input("请输入一个正整数 m:"))    #输入正整数 m while m&gt;0:     n = ②    #取 m 的右侧第一位数     print(n,end=")    #输出 n 不换行     m = ③    #去掉右侧第一位数</pre>
<p>模拟二十二</p>	
<p>防偷窥密码锁：该密码锁设计比较人性化，输入的密码中只要包含正确密码（需要连续，比如，密码设置为“202312”，当输入“4568348720231224564”，也可以认为输入正确），就可打开，但是如果连续3次输入都错误就会被锁定。</p> <p>用Python编辑器打开“D:\NetStuOp\70\70\663”文件夹下的文件“a3.py”，请在序号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre>password="202312"    #设初始密码为 202312 n=0 while n&lt;3:     yours=input("请输入密码：")     n+=1     if password ① yours:         print("欢迎回家")         ②    #跳出循环     ③ :         print("密码错误") if n ④ 3:     print("错误 3 次，请稍后再试")</pre>
<p>递归法打印杨辉三角形：杨辉三角，是二项式系数在三角形中的一种几何排列，中国南宋数学家杨辉1261年所著的《详解九章算法》一书中出现。杨辉三角的特点是每个数字等于上一行的左右两个数字之和。即第n+1行的第i个数等于第n行的第i-1个数和第i个数之和。</p> <p>用Python编辑器打开“D:\NetStuOp\70\70\661”文件夹下的文件“yh.py”，请在序号处填写正确的代码，且不改变源程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>  <pre>请输入杨辉三角形的行数：10 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1 1 8 28 56 70 56 28 8 1 1 9 36 84 126 126 84 36 9 1</pre>	<pre>def yanghui(x, y):     if y == ① or y == x:         return 1     else:         z = yanghui(②, y-1)+yanghui(x-1, ③)    # 第 x 行第 y 个数字等于第 x-1 行的第 y-1 个数和第 y 个数之和。         return z  n = int(input("请输入杨辉三角形的行数：")) for i in range(1, ④):    # 根据输入的行数打印杨辉三角形     for j in range(1, i+1):         print(yanghui(i, j), end=" ")    #调用函数，打印第 i 行的数     print()</pre>

<p><b>模拟二十三</b></p>	
<p>素数是指在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数。例如 23，它只有 1 和 23 两个因数。所以 23 就是一个素数，<b>@</b>文数是指正读和反读是一样的数字，例如 12321，它的正反读都是一样的，那么 12321 就是一个回文数。而 123 则不是回文数。设计 Python 程序查找[100,1900]范围内的所有回文素数。</p> <p>用 Python 编辑器打开“考生文件夹\655”文件夹下的文件“回文素数.py”，请在序号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre>import math def prime(n):     for i in range(2,int(math.sqrt(n))+1):         if ①_____:             return False     ②_____ def rev(n):     t=0     while ③_____:         t=t*10+n%10         n=n//10     return t for i in range(100,1001):     if rev(i)==i:         if ④_____:             print(i,end=" ")</pre>
<p>输入任意一个正整数 n，编程实现:输出由 1 到 n(包括 n)按顺序组合成的新数 123...n，例如输入 n 为 9，则输出新数为 123456789。用 Python 编辑器打开“考生文件夹\660”文件夹下的文件“a2.py”，请在序号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre>n= ① (input('请输入自然数 n:')) new="" for i in range(1, ② ):     ③ =new+ ④      #生成新的数 print(new)</pre>
<p><b>模拟二十四</b></p>	
<p>角谷猜想，是指对于任意一个正整数，如果是奇数，则乘 3 加 1，如果是偶数，则整除 2，得到的结果再按照上述规则重复处理，最终总能够得到 1。如，假定初始整数为 5，计算过程分别为 5、16、8、4、2、1，变化次数则为 5 次。请编程实现:输入两个正整数，输出该范围内角谷猜想变化次数最多的数及其变化次数，如果输入的第 1 个数大于第 2 个数，则交换。如输入 20 和 10，则计算出 10 到 20(含)之间变化次数最多的数是 18。变化次数为 20。在 Python 编辑器中打开“考生文件夹\656”下的“角谷猜想.py”文件，按下列要求进行操作并保存。注，程序修改时，请把题目序号删除，不能删除注释语句。</p>	<pre>n1 = int(input("请输入第一个数: ")) n2 = int(input("请输入第二个数: ")) if n1 &gt; n2:     ① maxStep = 0 maxNum = n1 for i in range(n1,②):     N = i     stepCount = 1     while N!= 1:         stepCount += 1         if ③:             N = N // 2         else:             N = N * 3 + 1     if stepCount &gt; maxStep:         maxStep = ④         maxNum = i print(maxNum) print(maxStep)</pre>
<p>回文数判断。如果一个正整数 n，若它的各位数字反向排列所得自然数 n1 与它本身相等，则 n 为回文数。例如，12321 为一回文数。编程实现:输入任意一个正整数 n，判断它是否为回文数。</p> <p>用 Python 编辑器打开“考生文件夹\659”文件夹下的文件“a1.py”，请在序</p>	<pre>def rev(x):     m=x     n1=0     while ① :</pre>

<p>号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。本题评析:(共 10 分得 0 分)</p>	<pre> n1=n1*10+ ② x=x//10 if m ③ n1:     return "是回文数。" else:     return "不是回文数。"  n=int(input("请输入任意正整数: ")) print( n, ④ ) </pre>
<p>模拟二十五</p>	
<p>小华设计了一个简易的自助点餐系统，功能是可以选择套餐 A 或套餐 B，其中套餐 B 还可以选择饮料，根据用户选择的套餐以及套餐的数量，计算所需支付的金额。</p> <p>用 Python 编辑器打开“考生文件夹\662”文件夹下的文件“dc.py”，请在序号处填写正确的代码，且不改变源程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre> ① food_A():     print("红烧肉")     print("炖鸡蛋")     print("青菜香菇")     return 10  def ②(drink):     print("土豆牛肉")     print("番茄鸡蛋")     print("白菜")     print(drink)     return 13  print("---欢迎您使用自助点餐系统---") food=input("请选择套餐 (A 或 B) ") if food=='A':     price=food_A()     num=int(input('您需用几份: '))     print("请支付", ③*num, "元，祝您用餐愉快! ") else:     dri=input("请选择一种饮料 (酸奶/冰红茶/可乐) ")     food_B(④)     num = int(input('您需用几份: '))     print("请支付", food_B(dri)*num, "元，祝您用餐愉快!") ) </pre>
<p>果实自动筛选，随着科技的发展，越来越多的无人化农业采摘机器人可以轻松实现自动采摘，分级封装。为此设计一个程序来解决“果实筛选并计重”的问题，筛选出 10 个大于 35 的整数存入列表 a 中，并计算出它们的和。说明。通过随机产生[30,40]的整数来模拟单个果实的重量。</p> <p>用 Python 编辑器打开“考生文件夹\664”文件夹下的文件“a4.py”，请在序号处填写正确的代码，且不改变原程序的结构，并把序号删除，调试完成后保存文件并关闭应用软件。</p>	<pre> from random ① randint #导入模块 a=list() #初始化列表 a n=0 #n 为个数 while ② &lt;11:     x= ③ (30,40) #产生随机整数     if x&gt;35:         n=n+1         a.append(x) print(a,④(a),sep="\n") </pre>