



朝阳区线上课堂·高中信息技术

程序设计题的分析与解答

北京中学 赵腾任

计算圆周率 π

- 人们创造了上百种方法求 π 的值。格雷戈里和莱布尼茨发现了下面的公式：
- 莱布尼茨级数： $\pi/4=1-1/3+1/5-1/7+1/9\ldots\ldots$
- 则： $\pi=(1-1/3+1/5-1/7+1/9\ldots\ldots)*4$
- 要求：精确到十万分之一，结果保留5位小数。

计算100以内奇数和

- 计算数的和：

把一系列数相加：累加

- 100以内的奇数：100以内的奇数有哪些，如何得到？
- 隐含的要求：计算的结果，需要输出出来。

计算100以内奇数和

- 计算数的和：存放→变量sum→sum=0
累加→次数固定→for循环
- 100以内的奇数：range(1,100,2)
- 输出计算的结果：print()

计算100以内奇数和

```
sum=0
for i in range(1,100,2):
    sum=sum+i
print(sum)
```

运行结果:

2500

计算100以内奇数和

```
sum=0
for i in range(1,100,2):
    sum=sum+i
print("1+3+5+...+99=",sum)
```

添加了说明文本后，运行结果为：

```
1+3+5+...+99= 2500
```

进阶：计算： $1 - 1/3 + 1/5 - \dots - 1/99$

• 与 “ $1+3+5+\dots+99$ ” 相比，发生了哪些变化：

1. 相加的数，由奇数变成了奇数的倒数： $n \rightarrow 1/n$

2. 相邻的项，符号发生了变化：

$$(+1) * (1/n), (-1) * (1/(n+2)), \dots$$

进阶：计算： $1-1/3+1/5-.....+1/99$

```
#计算：1-1/3+1/5-...+1/99=?
```

```
sum=0
```

```
m=1
```

```
for i in range(1,100,2):
```

```
    r=1/i
```

```
    sum=sum+m*r
```

```
    m=-m
```

```
print("1-1/3+1/5-...+1/99=", sum)
```

计算圆周率 π

- 比较：
- $1 - 1/3 + 1/5 - \dots + 1/99 = ?$
- $\pi = (1 - 1/3 + 1/5 - 1/7 + 1/9 \dots) * 4$
- 要求：精确到十万分之一，结果保留5位小数。

计算圆周率 π

- 比较：

- $1 - 1/3 + 1/5 - \dots + 1/99$

数列是确定项，可以使用for循环

- $\pi = (1 - 1/3 + 1/5 - 1/7 + 1/9 \dots) * 4$

数列是不确定项，需要随时根据要求进行判断

计算圆周率 π

- $\pi = (1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots) * 4$

要求：精确到十万分之一

也就是： $(1/i) * 4$ 小于十万分之一

当 $(1/i) * 4 < 0.00001$ 时，不再往下进行

计算圆周率 $\pi=(1-1/3+1/5-1/7+1/9\dots)*4$ ，精确到十万分之一

#用莱布尼茨级数计算 π ，精确到十万分之一

```
sum=0
```

```
m=1
```

```
i=1
```

```
while True:
```

```
    r=1/i
```

```
    sum=sum+m*r
```

```
    if r*4<0.00001:
```

```
        break
```

```
    m=-m
```

```
    i=i+2
```

```
pi=sum*4
```

```
print("用莱布尼茨级数计算 $\pi$ ，精确到十万分之一， $\pi=$ ",pi)
```

用莱布尼茨级数计算 π ，精确到十万分之一， $\pi = 3.141597653564762$

计算圆周率 $\pi=(1-1/3+1/5-1/7+1/9\ldots)*4$ ，精确到十万分之一，结果保留5位小数。

- 保留n为小数的问题，这里我们介绍一种方法
- 例如：3.123456789，保留3位小数
- $3.123456789*10^3\rightarrow 3123.456789$
- $\text{int}(3123.456789)/10^3\rightarrow 3.123$
- 其实还可能会涉及到四舍五入问题，思路并不难，请同学请进行探索。

#用莱布尼茨级数计算 π ，精确到十万分之一，保留5位小数

```
sum=0
```

```
m=1
```

```
i=1
```

```
while True:
```

```
    r=1/i
```

```
    sum=sum+m*r
```

```
    if r*4<0.00001:
```

```
        break
```

```
    m=-m
```

```
    i=i+2
```

```
pi=sum*4
```

```
pi=int(pi*100000)/100000
```

```
print("用莱布尼茨级数计算圆周率， $\pi$ =",pi)
```

折纸问题：

- 用一张厚度是0.1毫米的纸不停对折多少次，厚度可以达到珠峰高度8848米？
- 提示：简单起见，可以让纸的厚度和山的高度都乘以10、单位统一为毫米，则纸的厚度为1，珠峰的高度为88480000。



折纸问题：

首先对这个问题进行分解：

- 计算折叠后的厚度
- 判断折叠后是否达到设定厚度
- 重复折叠
- 统计折叠次数
- 输出结果

折纸问题：

- 计算折叠后的厚度：

折叠后厚度 $\leftarrow$ 折叠前厚度 $\times 2$

折叠次数为 n ，则折叠后的厚度为 2^n



折纸问题：

- 判断折叠后是否达到设定厚度

如果：厚度 $\geq$ 目标值，停止对折

如果：厚度 $<$ 目标值，继续对折



折纸问题：

- 重复折叠

折叠多少次才能超过珠峰高度

是“不确定”的，需要进行计算和判断

这里可以使用while循环

折纸问题：

- 统计折叠次数

次数 n ：0、1、2、3、4、5、6.....

需要对折叠的次数进行累加： $n=n+1$



折纸问题：

- 输出结果

输出：折叠的次数

输出：达到的高度

为方便计算，我们暂时将单位统一为毫米，最后我们要将它转换为米。

#纸的厚度为0.1毫米，将纸进行对折，多少次能达到珠峰高度
#为便于计算，将单位统一为毫米，并都乘以10

t=1 #纸的初始厚度t为：1 (mm)

h=88480000 #珠峰高度h为：88480000 (mm)

n=0 #设置折叠次数n初始值为：0

while t<h: #当折纸的厚度小于设定值，则进行循环

 n+=1 #次数加一

 t=t*2 #每折叠一次、厚度乘2

t=**int**(t/10000)

print('折叠',n,'次',',',end='')

print('厚度达到：',t,'米，超过珠峰高度。')

- 
- 
- 用计算机解决问题的一般过程：

分析问题→设计方案→编程调试→解决问题

- 编程解决问题的基本步骤：

分析问题→设计算法→编程实现与调试→保存文件、调试运行程序



小结：

- 首先，细读文本，对问题有一个整体认识。
- 其次：提炼关键词，找出已知条件。
- 第三：确定求解目标，包括隐含的目标。
- 然后：确定已知与未知的关系。
- 最后：设计算法、编程、调试。



比如：根据输入的身高和体重值，编程计算体重指数。

- 身高和体重：有了这两个值，才能计算体重指数。
- 输入：运行程序时，从键盘输入。
- 体重指数：这是要求解的目标，要输出。
- 身高、体重，与BMI之间的关系：体重指数公式。
- 画出流程图，编写、调试程序。



感谢您的观看

北京市朝阳区教育研究中心 制作