

# Lua 语言编程专题研究

2018 年版

郝立丽 郝立柱

应用统计教育部重点实验室  
东北师范大学数学与统计学院

## 前 言

Lua 语言是简单而小巧的解释性语言, 主要用于硬件的高层抽象、动态结构、无冗余、简易的测试与调试. 它具有的概念不多, 但是每个概念都很有用, 功能很强大. 它的简易性使得 Lua 非常容易学习, 众多的函数可以很方便进行文本处理. Lua 只有一种数据结构, 那就是表, 它非常强大, 表可以当数组用, 也可以当堆栈用, 甚至可以用其构造队列、链表等, 即使是图也很容易通过表来构造. Lua 是一种可扩展的语言, 可以很容易地和 C 语言衔接, 我们可以将其看成是应用领域的工具包. 它的移植性也很好, 全部代码几乎都是用标准 C 开发的.

应用统计教育部重点实验室为了能够适应海量数据处理, 建议在籍学生掌握 Lua 语言, 并以 Lua 语言为根基, 扩展到科研的各个领域.

本书从编程的角度出发, 共计十二个专题讨论, 详细探究了基本编程技巧、基本编程技能及其基本编程方法.

全文使用 A4 纸张按照双栏排版, 学习时可以采用打印后, 慢慢仔细研究编程规范和基本编程习惯. 本书提供全部的程序文件以及相关辅助文件, 这便于调试, 理解程序运行过程中的每一步执行状态.

关于作者情况简述如下: 郝立柱和郝立丽分别工作于东北师范大学数学与统计学院和东北林业大学经济管理学院, 都是应用统计教育部重点实验室机器学习课题组成员, 郝立柱的电子邮箱: HSystem@163.com, 如果读者有任何建议和意见欢迎来函批评指正.

郝立柱 郝立丽

2018 年 10 月 28 日

## 目 录

第 1 章 顺序结构 .....	1
1.1 表达式 .....	1
1.2 输入与输出 .....	1
1.3 简单计算 .....	1
1.4 计算繁分式 .....	1
1.5 函数计算 .....	1
1.6 求方程的根 .....	1
1.7 求二元一次方程组的解 .....	1
1.8 计算三角形面积 .....	2
1.9 整数除法余数问题 .....	2
1.10 交换 a,b 之值 .....	2
1.11 整数的数位个数 .....	2
1.12 习题 .....	2
第 2 章 选择结构 .....	3
2.1 判断语句格式 .....	3
2.2 找出正整数 a,b 之中的最大者 .....	3
2.3 找出正整数 a,b,c 三者中的最大者 .....	3
2.4 将 a,b,c 按从小到大排序 .....	3
2.5 判断谁大 .....	4
2.6 各个月份的天数 .....	4
2.7 判断是否是闰年 .....	4
2.8 习题 .....	4
第 3 章 循环结构 .....	5
3.1 while 格式 .....	5
3.2 repeat 格式 .....	5
3.3 数字型 for 格式 .....	5
3.4 泛型 for 格式 .....	5
3.5 3X+1 问题 .....	5
3.6 求正整数 (a,b) 的最大公约数 .....	5
3.7 二分法求方程的根 .....	5
3.8 $S = 1 + 2 + 3 + \dots + 100$ .....	5
3.9 $S = 1 * 2 + 2 * 3 + \dots + 99 * 100$ .....	6
3.10 $S = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{100}$ .....	6
3.11 $S = \frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} \dots + \frac{1}{100*101}$ .....	6
3.12 $S = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \dots + \frac{1}{99} - \frac{1}{100}$ .....	6

3.13 素数 (质数) 判断 .....	6
3.14 遍历数组的全部元素 .....	7
3.15 遍历对象的全部元素 .....	7
3.16 完全平方数 .....	7
3.17 习题 .....	8
第 4 章 二重循环 .....	9
4.1 图形打印 .....	9
4.1.1 实心矩形 .....	9
4.1.2 左三角实心图形 .....	9
4.1.3 右三角实心图形 .....	9
4.1.4 上三角实心图形 .....	9
4.1.5 字母实心图形 .....	9
4.1.6 数字实心图形 .....	10
4.1.7 连续数字实心图形 .....	10
4.2 计算三阶水仙花数 .....	10
4.3 暴力搜索: 百钱买百鸡 .....	10
4.4 级数求和 .....	10
4.5 变长序列求和 .....	11
4.6 简单运动过程 .....	11
4.7 习题 .....	12
第 5 章 一维数组 .....	13
5.1 数组的定义及初始化 .....	13
5.2 替换数组中的元素 .....	13
5.3 去掉数组中的某些元素 .....	13
5.4 数组元素查询 .....	13
5.5 数组元素插入 .....	14
5.6 生成菲波纳切数列 .....	14
5.7 分离十进制整数 a 的每一位数字 .....	14
5.8 成绩均值与方差 .....	14
5.9 集合的交集 .....	14
5.10 集合的并集 .....	15
5.11 集合的差集 (或补集) .....	15
5.12 习题 .....	16
第 6 章 函数调用 .....	17
6.1 函数定义 .....	17
6.2 最大公因子函数 .....	17
6.3 最小质因数函数 .....	17
6.4 正整数的质因数分解 .....	17

6.5 随机数发生器函数.....	18
6.6 组合函数.....	18
6.7 计算今天星期几.....	18
6.8 字符串数组的全连接.....	19
6.9 复杂计数器.....	19
6.10 习题.....	20
第 7 章 二维数组.....	21
7.1 二维数组的定义及初始化.....	21
7.2 矩阵加法.....	21
7.3 矩阵的消法变换.....	21
7.4 矩阵的转置变换.....	21
7.5 学生成绩.....	22
7.6 习题.....	22
第 8 章 堆栈技术.....	23
8.1 堆栈的一般形式.....	23
8.2 十进制转换为二进制.....	23
8.3 回文数.....	23
8.4 质因数分解.....	23
8.5 数组数据翻转.....	23
8.6 字符串反序.....	23
8.7 复杂连分数计算.....	24
8.8 习题.....	24
第 9 章 函数递归.....	25
9.1 十进制转换为二进制.....	25
9.2 菲波纳切数列的递归计算.....	25
9.3 递归计算杨辉三角.....	25
9.4 伞型序递归.....	25
9.5 利用递归求最大公因子.....	25
9.6 对策局势的确定.....	25
9.7 利用递归实现连分数计算.....	26
9.8 递归求根.....	26
9.9 习题.....	26
第 10 章 文件操作.....	27
10.1 文件操作的定义.....	27
10.2 写文件.....	27
10.3 数组内容写文件.....	27
10.4 读文件并存于数组中.....	27
10.5 将二维数组内容写入文件中.....	27

10.6 从文件中读入数据到二维数组中 .....	27
10.7 不定长数据的存储 .....	28
10.8 不定长数据的读取 .....	28
10.9 文件合并 .....	28
10.10 文件内容比较 .....	29
10.11 打印图形 .....	29
10.12 习题 .....	29
第 11 章 综合例题 .....	31
11.1 利润提成的计算 .....	31
11.2 约瑟夫问题 .....	31
11.3 狼找兔子 .....	32
11.4 包含 3 的数的个数 .....	32
11.5 表达式实例 .....	32
习题参考答案 .....	33
第 1 章答案 .....	33
第 2 章答案 .....	33
第 3 章答案 .....	33
第 4 章答案 .....	34
第 5 章答案 .....	35
第 6 章答案 .....	36
第 7 章答案 .....	36
第 8 章答案 .....	37
第 9 章答案 .....	38
第 10 章答案 .....	40
附录 1 库函数列表 .....	41

# 第一章 顺序结构

## §1.1 表达式

由四则运算符加减乘除以及括号和函数构成的式子, 就叫做表达式. 例如:  $12*13+50$

## §1.2 输入与输出

从键盘读入一个数, 将这个数乘以 3 然后输出到屏幕上.

```
程序名称: C01_010.lua
g=io.read("*number")
print(3*g)
输入: 5
输出: 15
```

程序运行时系统会暂时停下来, 等待从键盘上获得用户提供的数据, 比如输入: 5 然后按回车.

## §1.3 简单计算

计算:  $124*126$

```
程序名称: C01_020.lua
print(124*126)
结果为: 15624
```

从键盘输入两个数, 计算并输出其乘积.

```
程序名称: C01_030.lua
g,h=io.read("*number","*number")
print(g*h)
输入: 3 4
输出: 12
运行时, 输入两个数, 用空格隔开.
```

## §1.4 计算繁分式

$$\frac{11}{12 + \frac{3}{23}}$$

```
程序名称: C01_040.lua
print(11/(12+3/23))
结果为: 0.90681003584229
```

还可以这样写程序:

```
程序名称: C01_050.lua
x=3/23
y=11/(12+x)
print(y)
```

输出时限制输出长度: (保留小数点后 3 位)

```
程序名称: C01_060.lua
print(string.format("%.3f", 11/(12+3/23)))
结果为: 0.907
```

不能  
简化

## §1.5 函数计算

求值  $\sin(28.5^\circ) + \cos(12.3^\circ) + \sqrt{2.4}$

```
程序名称: C01_070.lua
x=math.sin(28.5/180*math.pi)
y=math.cos(12.3/180*math.pi)
print(x + y + math.sqrt(2.4))
结果为: 3.0033976731778
```

## §1.6 求方程的根

已知方程有两个不相同的实根, 求方程的根.

$$x^2 - 5x + 3 = 0$$

```
程序名称: C01_080.lua
a=1; b=-5; c=3
x1=(-b+math.sqrt(b*b-4*a*c))/(2*a)
x2=(-b-math.sqrt(b*b-4*a*c))/(2*a)
print(string.format("x1=%f", x1))
print(string.format("x2=%f", x2))
结果为:
x1=4.302776
x2=0.697224
```

韦达定理

## §1.7 求二元一次方程组的解

计算二元一次方程组的解:

$$\begin{cases} 3x_1 + 4x_2 = 5 \\ 2x_1 + x_2 = 3 \end{cases}$$

其解为:

$$x_1 = \frac{\begin{vmatrix} c_1 & a_{12} \\ c_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \quad x_2 = \frac{\begin{vmatrix} a_{11} & c_1 \\ a_{21} & c_2 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}$$

```

程序名称:C01_090.lua
a11=3; a12=4; c1=5;
a21=2; a22=1; c2=3;
x1=(c1*a22-c2*a12)/(a11*a22-a12*a21)
x2=(a11*c2-a21*c1)/(a11*a22-a12*a21)
print("x1=..x1")
print("x2=..x2")
结果为:
x1=1.4
x2=0.2

```

### §1.8 计算三角形面积

已知三角形的两边及其夹角, 求三角形面积, 其中  $a=3, b=4, \angle C=30^\circ$ . 面积

$$S = \frac{1}{2}ab \sin C$$

```

程序名称:C01_100.lua
a=3; b=4;
C=30/180*math.pi
s=0.5*a*b*math.sin(C)
print("s"..s)
结果为:s=3

```

### §1.9 整数除法余数问题

问 1234 除以 19 余数是多少?

```

程序名称:C01_110.lua
x=1234%19
print(x)
结果为:18

```

### §1.10 交换 a,b 之值

原来是  $a=3, b=4$  交换 a,b 之内容变为  $a=4, b=3$ .

```

程序名称:C01_120.lua
a=3; b=4;
print(a,b)
a,b=b,a
print(a,b)
结果为:
3 4
4 3

```

### §1.11 整数的数位个数

问任意正整数  $x$  是由十进制几位数字组成?

```

程序名称:C01_130.lua
a=io.read("*number")
s=tostring(a)
print(#s)

```

输入为:123321  
结果为:6

### §1.12 习题

#### 1. 函数计算

$$e^{\sin(18^\circ)} \log_{10} 42 + e^{12 \ln(13)}$$

#### 2. 计算繁分式

$$12 + \frac{11}{13 + \frac{23}{5}}$$

#### 3. 求三元一次方程组的解

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ 2x_1 + 2x_2 + x_3 = 3 \\ 3x_1 + 4x_2 + 5x_3 = 5 \end{cases}$$

#### 4. 计算三角形面积

已知三角形的三边的长度, 求三角形面积, 其中  $a=3, b=4, c=6$ . 面积公式

$$S = \sqrt{p(p-a)(p-b)(p-c)}$$

其中  $p$  为半周长, 即  $p = \frac{1}{2}(a+b+c)$ .

#### 5. 整数除法余数问题, 问:

1234 除以 11 的余数乘以

1234 除以 17 的余数乘以

1234 除以 19 的余数等于几?



## 第二章 选择结构

### §2.1 判断语句格式

第一种形式:

```
if 条件 then
    程序块1
end
```

第二种形式:

```
if 条件 then
    程序块1
else
    程序块2
end
```

第三种形式:

```
if 条件 then
    程序块1
elseif 条件 then
    程序块2
elseif 条件 then
    程序块3
else
    程序块4
end
```

### §2.2 找出正整数 a,b 之中的最大者

比如: 输入 3,6 输出 6

```
程序名称:C02_010.lua
a,b=io.read("*number","*number")
if a<=b then
    print("max"..b)
else
    print("max"..a)
end
输入为:3 6
输出为:max=6
```

还可以用函数求出最大值。

```
程序名称:C02_020.lua
a,b=io.read("*number","*number")
c=math.max(a,b)
print("max"..c)
输入为:3 6
输出为:max=6
```

### §2.3 找出正整数 a,b,c 三者中的最大者

比如: 输入 3,6,2 输出 6

```
程序名称:C02_030.lua
a,b,c=io.read("*number","*number","*number")
if a<=b then
    if b<=c then
        io.write("max=",c)
    else
        io.write("max=",b)
    end
else
    if a<=c then
        io.write("max=",c)
    else
        io.write("max=",a)
    end
end
输入为:3 6 2
输出为:max=6
```

还可以用函数求出最大值。

```
程序名称:C02_040.lua
a,b,c=io.read("*number","*number","*number")
d=math.max(a,b,c)
io.write("max=",d)
输入为:3 6 2
输出为:max=6
```

write 与 print 的区别: 首先,write 在输出时不会添加像制表符或回车这样的额外字符; 其次,write 使用当前输出文件, 而 print 总是使用标准输出; 最后,print 会自动调用其参数的 tostring() 方法, 因此它还能显示 table、函数和 nil.

### §2.4 将 a,b,c 按从小到大排序

比如: 输入 4,1,8 输出 1,4,8

```
程序名称:C02_050.lua
a,b,c=io.read("*number","*number","*number")
if a>b then a,b=b,a end
if b>c then b,c=c,b end
if a>b then a,b=b,a end
print(a,b,c)
输入为:4 1 8
输出为:1 4 8
```

还可以用函数进行排序。

```
程序名称:C02_060.lua
a={io.read("*number","*number","*number")}
table.sort(a)
print(table.concat(a,"\t"))
```

输入为:4 1 8

输出为:1 4 8

从大到小排序呢?

比如:输入 3,6,2 输出 6,3,2

程序名称:C02\_070.lua

```
a={io.read("*number","*number","*number")}
table.sort(a,function(a,b) return a>b end)
print(table.concat(a,"\t"))
```

输入为:3 6 2

输出为:6 3 2

## §2.5 判断谁大

判断  $\sin(x)$  与  $x - (1/6)x^3$  谁大? 其中  $x = 1.2345$

程序名称:C02\_080.lua

```
x=io.read("*number")
if math.sin(x)>x-1/6*x^3 then
    print("sin(x) 大")
else
    print("x-1/6*x^3 大")
end
```

输入为:1.2345

输出为:sin(x) 大

## §2.6 各个月份的天数

键盘输入月份,系统输出该月份所含有的天数.(不考虑闰年)

使用 if 的方式实现.

程序名称:C02\_090.lua

```
x=io.read("*number")
if x==4 or x==6 or x==9 or x==11 then
    io.write("day=30")
elseif x==2 then
    io.write("day=28")
else
    io.write("day=31")
end
```

输入为:9

输出为:day=30

使用 table 的方式实现.

程序名称:C02\_100.lua

```
day={31,28,31,30,31,30,31,31,30,31,30,31}
x=io.read("*number")
print(day[x])
```

输入为:8

输出为:31

## §2.7 判断是否是闰年

例如:

1900 --> 不是

1904 --> 是

程序名称:C02\_110.lua

```
i=1900
if i%400==0 or (i%100~=0 and i%4==0) then
    print("是")
else
    print("不是")
end
```

## §2.8 习题

1. 找出正整数 a,b,c,d 四者中的最大者.
2. 键盘输入四个正整数,将这四个正整数 a,b,c,d 按从大到小排序.
3. 当  $x = 1.2345$  时,判断  $\tan(x)$  与  $x - (1/3)x^3$  谁大?
4. 符号函数

$$\begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$$

键盘输入一个整数,输出其符号,即 -1,0,1.

5. 求方程  $ax^2 + bx + c = 0$  的根,其中  $a, b, c$  为实数,并由键盘输入.

无实数根时,输出“无解”;

有单一实根时,输出“ $x_1 = x_2 =$ ”并输出那个解;

若有两个不同的实根时,输出每一个实根.

## 第三章 循环结构

### §3.1 while 格式

先判断条件, 为真时执行循环体, 否则退出循环体.

```
while 条件 do
  循环体
end
```

### §3.2 repeat 格式

先执行循环体, 然后判断条件, 为真时退出循环体.

```
repeat
  循环体
until 条件
```

### §3.3 数字型 for 格式

```
for 循环变量=初值,终值,增量 do
  循环体
end
```

增量可以不写, 默认为 1. 若不想给循环设置上限, 则上限可以用 math.huge.

### §3.4 泛型 for 格式

```
for i,v in ipairs(a) do --数组a
  循环体
end
-- 其中i为数组索引,v为数组元素值

for k in pairs(t) do --表t
  循环体
end
-- 其中k为表t的key
```

注意: 循环变量是局部变量, 循环体结束即作废.

### §3.5 3X+1 问题

从键盘输入一个正整数  $x$ , 然后按照下面的规则, 进行迭代, 直至出现 1 为止.

$$f(x) = \begin{cases} 3x+1 & \text{当 } x \text{ 为奇数时} \\ x/2 & \text{当 } x \text{ 为偶数时} \end{cases}$$

例如: 输入 6, 系统输出

6,3,10,5,16,8,4,2,1

程序名称:C03\_010.lua

```
x=io.read("*number")
while x~=1 do
  io.write(x..",")
  if x%2==0 then x=x/2 else x=3*x+1 end
end
io.write(x)
```

奇偶判别

### §3.6 求正整数 (a,b) 的最大公约数

使用辗转相除法求最大公约数, 即若  $a > b$ , 则  $a$  除以  $b$  后的余数存入  $a$  中, 否则  $b$  除以  $a$  后的余数存入  $b$  中, 反复迭代直至  $a, b$  中有一个为零时停止.

程序名称:C03\_020.lua

```
a,b=io.read("*number","*number")
while a>0 and b>0 do
  if a>b then a=a%b else b=b%a end
end
print(a+b)
输入:48 54
输出:6
```

### §3.7 二分法求方程的根

已知函数

$$f(x) = x^3 - 5x + 1, \quad f(0) > 0, \quad f(1) < 0$$

由于函数  $f(x)$  连续, 可见在  $(0,1)$  区间上必然存在一个根, 求其根. 精确到小数点后 3 位, 即 0.001.

程序名称:C03\_030.lua

```
a=0; b=1;
while b-a>0.001 do
  c=(a+b)/2
  if c^3-5*c+1>0 then a=c else b=c end
end
print(a)
```

结果为:0.201171875

### §3.8 $S = 1 + 2 + 3 + \dots + 100$

1. 产生 1,2,3,4,..., 方法是  $i = i + 1$ ;

2. 累加  $1, 1+2, 1+2+3, \dots$ , 方法是  $s = s + i$ ;

使用 while 循环实现

程序名称:C03\_040.lua

```
s=0; i=1
while i<=100 do
    s=s+i
    i=i+1
end
print(s)
结果为:5050
```

使用 for 循环实现

程序名称:C03\_050.lua

```
s=0
for i=1,100,1 do
    s=s+i
end
print(s)
```

$$\S 3.9 \quad S = 1 * 2 + 2 * 3 + \dots + 99 * 100$$

1. 产生  $1, 2, 3, 4, \dots$ , 方法是  $i = i + 1$ ;

2. 计算中间结果  $1 * 2, 2 * 3, \dots$ , 方法是  $t = i * (i + 1)$ ;

2. 累加  $s = 1 * 2, s = s + 2 * 3, s = s + 3 * 4, \dots$ , 方法是  $s = s + t$ ;

使用 while 循环实现

程序名称:C03\_060.lua

```
s=0; i=1
while i<=99 do
    t=i*(i+1)
    s=s+t
    i=i+1
end
print(s)
结果为:333300
```

使用 for 循环实现

程序名称:C03\_070.lua

```
s=0
for i=1,99 do
    s=s+i*(i+1)
end
print(s)
```

$$\S 3.10 \quad S = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} \dots + \frac{1}{100}$$

程序名称:C03\_080.lua

```
s=0
for i=1,100 do
    s=s+1/i
end
print(s)
结果为:5.1873775176396
```

$$\S 3.11 \quad S = \frac{1}{1*2} + \frac{1}{2*3} + \frac{1}{3*4} \dots + \frac{1}{100*101}$$

程序名称:C03\_090.lua

```
s=0
for i=1,100 do
    s=s+1/(i*(i+1))
end
print(s)
结果为:0.99009900990099
```

$$\S 3.12 \quad S = \frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} \dots + \frac{1}{99} - \frac{1}{100}$$

程序名称:C03\_100.lua

```
s=0; k=1
for i=1,100 do
    s=s+k/i;
    k=-k;
end
print(s)
结果为:0.6881721793102
```

还有一种处理交替符号的方法, 这种方法还可以进一步推广为处理任意具有周期特征的符号现象。

程序名称:C03\_110.lua

```
s=0; k=0
for i=1,100 do
    if k==0 then
        t=1/i
    else
        t=-1/i
    end
    s=s+t;
    k=k+1;
    if k>=2 then k=k-2 end
end
print(s)
```

作用类似于开关, 可任意更改换号周期

### §3.13 素数 (质数) 判断

从键盘输入一个大于 1 的正整数, 编程判断其是否为素数。

比如:

输入 12 系统输出 No  
输入 91 系统输出 No  
输入 101 系统输出 Yes

程序名称:C03\_120.lua

x=io.read("\*number")

flag=0 -- 开关变量(关)

```
for i=2,x-1 do
    if x%i==0 then flag=1; break end
end
```

if flag==1 then

print("合数")

else

print("质数")

end

### §3.14 遍历数组的全部元素

程序名称:C03\_130.lua

aa={"星期一","星期二","星期三"},}

```
for i,v in ipairs(aa) do
    print(v)
end
```

print("另一种形式:")

```
for i,v in ipairs(aa) do
    print(i,aa[i])
end
```

end

结果为:

星期一

星期二

星期三

另一种形式:

1 星期一

2 星期二

3 星期三

i表示数组元素位置(地址),v表示对应位置的元素

### §3.15 遍历对象的全部元素

程序名称:C03\_140.lua

aa={x="星期一",y="星期二"}

aa.z="星期三"

```
for k in pairs(aa) do
```

print(k)

end

print("另一种形式:")

```
for k in pairs(aa) do
```

print(k,aa[k])

end

结果为:

y

x

z

另一种形式:

y 星期二

x 星期一

z 星期三

### §3.16 完全平方数

例1: 对于不超过10万的正整数,它加上100后是一个完全平方数,再加上168又是一个完全平方数,输出满足该条件的所有数.

简单分析: 假设这个正整数为x,则它满足 $p = x + 100 = r^2$ 和 $q = x + 100 + 168 = s^2$ ,且r和s均为正整数.

程序名称:C03\_150.lua

```
for i=1,100000 do
```

p=i+100; r=p^0.5; r=r-r%1

q=p+168; s=q^0.5; s=s-s%1

if p==r^2 and q==s^2 then

print(i,r,s,i+100,r^2,i+268,s^2)

end

end

例2: 有这样一个6位数,它本身是一个整数的平方,其高三位和低三位也分别是一个整数的平方,如 $225625 = 475^2$ , $225 = 15^2$ , $625 = 25^2$ ,求满足上述条件的所有6位数.

分析: 对于一个给定的六位数,首先判断它是否为一个整数的平方,如果它为一个整数的平方,则将它的高三位和低三位分别表示出来,再判断它的高三位和低三位是否分别为其它两个整数的平方,若是,则输出该数.

程序名称:C03\_160.lua

```
for i=100000,999999 do
```

k=i^0.5; k=k-k%1

if k^2==i then

k1=i%1000; k2=(i-k1)/1000

k=k1^0.5; k=k-k%1

if k^2==k1 then

k=k2^0.5; k=k-k%1

if k^2==k2 then

print(i)

end

end

end

end

## §3.17 习题

1. 求正整数 (a,b,c) 的最大公约数.

2. 二分法求方程的根

已知函数

$$f(x) = \sin(x) + 2x - 1, \quad f(0) < 0, f(1) > 0$$

由于函数  $f(x)$  连续, 可见在 (0,1) 区间上必然存在一个根, 求其根. 精确到小数点后 2 位, 即 0.01.

3. 计算数列和.

$$S = \frac{1}{1} + \frac{1}{2} - \frac{1}{3} - \frac{1}{4} \cdots - \frac{1}{99} - \frac{1}{100}$$

符号为  $++--++--\cdots$ , 先两个正数, 然后两个负数, 循环往复.

4. 余数问题 (中国剩余定理)

今有物不知其数, 三三数之剩二, 五五数之剩三, 七七数之剩二, 问物几何?

(提示: 从 1 到  $\infty$  逐一测试, 直到找到满足上述条件的数为止)

5. 青年歌手参加歌曲大奖赛, 有 10 个评委打分, 试编程求选手的平均得分 (去掉一个最高分和一个最低分).

(提示: 将 10 个分数存于一个数组中, 按从小到大的顺序对数组进行排序. 排序后的数组从第二个元素开始相加直到倒数第二位停止, 然后除以 (10-2) 即可.)

6. 青年歌手参加歌曲大奖赛, 有 20 个评委打分, 试编程求选手的平均得分 (去掉三个最高分和三个最低分).

## 第四章 二重循环

### §4.1 图形打印

#### §4.1.1 实心矩形

打印如下图形:

```
*****
*****
*****
*****
*****
程序名称:C04_010.lua
for i=1,5 do
    print("*****")
end
```

以上是一次打印一行,这种方式不够灵活,可以再分细,每一行中每次仅打印一个字符,也就是说再用一重循环,可以控制到每次打印一个字符的目的。

```
程序名称:C04_020.lua
for i=1,5 do
    for j=1,10 do
        io.write("*")
    end
    io.write("\n")
end
```

#### §4.1.2 左三角实心图形

打印如下图形:

```
*
**
***
****
*****
程序名称:C04_030.lua
for i=1,5 do
    for j=1,i do
        io.write("*")
    end
    io.write("\n")
end
```

#### §4.1.3 右三角实心图形

打印如下图形:

```
*
**
***
****
*****
```

本题在打印时,由于很难控制光标在屏幕上的位置,所以最简单的方法是通过打印一些空白字符来占位,从而实现对齐图形的目的。

```
程序名称:C04_040.lua
for i=1,5 do
    for j=1,5-i do
        io.write(" ")
    end
    for j=1,i do
        io.write("*")
    end
    io.write("\n")
end
```

注意:第二个循环语句不需要写j=6-i,5

#### §4.1.4 上三角实心图形

打印如下图形:

```
*
***
*****
*****
*****
*****
```

这种非对齐的打印不了

```
程序名称:C04_050.lua
for i=1,5 do
    for j=1,5-i do
        io.write(" ")
    end
    for j=1,2*i-1 do
        io.write("*")
    end
    io.write("\n")
end
```

#### §4.1.5 字母实心图形

打印如下图形:

```
a
bbb
ccccc
ddddddd
eeeeeeee
```

```
程序名称:C04_060.lua
ch=97
for i=1,5 do
    for j=1,5-i do io.write(" ") end
    for j=1,2*i-1 do
        io.write(string.char(ch))
    end
    ch=ch+1
    io.write("\n")
end
```

## §4.1.6 数字实心图形

打印如下图形:

```

1
123
12345
1234567
123456789

```

程序名称:C04\_070.lua

```

for i=1,5,1 do
  for j=1,5-i,1 do
    io.write(" ")
  end
  ch=1
  for j=1,2*i-1,1 do
    io.write(ch)
    ch=ch+1
  end
  io.write("\n")
end

```

数字也可以直接打印,无需  
用字符表示



## §4.1.7 连续数字实心图形

打印如下图形:

```

      1
     2 3 4
    5 6 7 8 9
   10 11 12 13 14 15 16
  17 18 19 20 21 22 23 24 25

```

程序名称:C04\_080.lua

```

k=0
for i=1,5 do
  for j=1,5-i do
    io.write("   ")
  end
  for j=1,2*i-1 do
    k=k+1
    io.write(string.format("%4d",k))
  end
  io.write("\n")
end

```

空格要

为了对齐

### ✓ §4.2 计算三阶水仙花数

一个三位数,若其各个数位上的数的立方和恰好等于该数本身,就称之为三阶水仙花数.例如:

$$153 = 1^3 + 5^3 + 3^3$$

求所有的三阶水仙花数.

程序名称:C04\_090.lua

```

for i=1,9 do
  for j=0,9 do
    for k=0,9 do
      m=i*i*i+j*j*j+k*k*k
      if i*100+j*10+k==m then
        io.write(i*100+j*10+k, " ")
      end
    end
  end
end

```

输出为:153 370 371 407

拆分个,十,百位。

## §4.3 暴力搜索:百钱买百鸡

鸡翁一值钱五,鸡母一值钱三,鸡雏三值钱一,百钱买百鸡,问鸡翁母雏各几何?

显然,鸡雏的个数一定是3的倍数,故令鸡雏从0搜索到99,鸡翁从0搜索到20,鸡母只需100-鸡翁-鸡雏即可.

程序名称:C04\_100.lua

```

for i=0,20 do
  for k=0,99,3 do
    j=100-i-k
    if j>=0 and i*5+j*3+k/3==100 then
      print(i,j,k)
    end
  end
end

```

输出为:

0	25	75
4	18	78
8	11	81
12	4	84

注意范围

## §4.4 级数求和

$$S = \frac{1}{1} + \frac{1}{1+2} + \frac{1}{1+2+3} + \cdots + \frac{1}{1+2+\cdots+100}$$

程序名称:C04\_110.lua

```

s=0
for i=1,100 do
  t=0
  for j=1,i do
    t=t+j
  end
  s=s+1/t
end

```



```
end
print(s)
输出为:1.980198019802
```

可储存中间结果作用很大

如果存储中间结果, 则程序的效率会更高. 通常情况, 对于能够提高程序效率的中间结果, 一般都加以利用.

```
程序名称:C04_120.lua
s=0; t=0
for i=1,100 do
    t=t+i;
    s=s+1/t;
end
print(s)
```

循环

#### §4.5 变长序列求和

求  $s=1+12+123+1234+\dots+12345678$  的值.

算法:

1. 用  $a$  表示每个数,  $s$  表示累加和;

2.  $a=a*10+i$ ,  $a*10$  表示把  $a$  每次放大 10 倍, 再加上个位数  $i$  ( $i$  为循环变量), 就是下一个数;

3.  $s=s+a$ , 即到目前为止所有数的累加和.

```
程序名称:C04_130.lua
a=0
s=0
for i=1,8 do
    a=a*10+i
    s=s+a
end
print(s)
结果为:13717416
```

#### §4.6 简单运动过程

一球从 100 米高度自由落下, 每次落地后反跳回原高度的一半; 再落下, 求它在第 10 次落地时, 共经过多少米? 第 10 次反弹多高?

分析: 每次落地后反弹回原高度的一半, 这些高度就形成一个如 100、50、25 …… 的等差数列, 公差为  $1/2=0.5$ , 所以通项公式为 100 乘以 0.5 的  $n$  次方, 第 10 次即  $n$  取 10 可求得 0.097656 米, 再将 100、50、25 …… 这 10 项加起来乘以 2 再减 100 (第一项是单程, 后面每次上下往返所以乘以 2) 得 299.609375 米.

程序名称:C04\_140.lua

```
a=100
s=-100
for i=1,10 do
    s=s+2*a
    a=a/2
end
print(s,a)
结果为:299.609375 0.09765625
```

100, 50, 25, 12.5, 6.25, 3.125, 1.5625, 0.78125, 0.390625, 0.1953125, 0.09765625

$S_1, S_2, S_3$



## §4.7 习题

1. 打印如下图形:

```
*****
****
***
**
*
```

2. 打印如下图形:

```
*****
****
***
**
*
```

3. 打印如下图形:

```
      *
    ***
  *****
*****
*****
*****
***
*
```

4. 打印如下图形:

```
aaaa*
aaa**
aa***
a****
```

5. 打印如下图形:

```
aaaa*aaaa
aaa***aaa
aa*****aa
a*****a
```

6. 打印如下图形:

```
      *      *
    ***    ***
  *****  *****
*****  *****
*****  *****
```

7. 打印如下图形:

```
      *      *
    ***    ***
  *****  *****
*****  *****
*****  *****
*****  *****
*****  *****
***      ***
*
```

8. 打印如下图形:

```
a
abc
abcde
abcdefg
abcdefghi
abcdefg
abcde
abc
a
```

9. 打印如下图形:

```
a
aba
abcba
abcdcba
abcdedcba
abcdcba
abcba
aba
a
```

10. 计算四阶水仙花数

一个四位数, 若其各个数位上的数的四次方和恰好等于该数本身, 就称之为四阶水仙花数. 例如:

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

求所有的四阶水仙花数.

11. 输出 100 以内的全部素数.(参见上节素数判断)

12. 暴力搜索满足方程  $3x + 5y = 98$  的全部正整数解.13. 级数求和:  $S = 1! + 2! + 3! + \cdots + 7!$

## 第五章 一维数组

切记：千万千万不要把一个数组直接赋值给另一个！

### §5.1 数组的定义及初始化

```
aa={}
bb={1,5,3,6,8,9,3,2,0,4}
cc={12.19,13.666,14.5}
dd={"abc","bcd","1200"};
```

### §5.2 替换数组中的元素

已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的偶数全部替换成 0, 要求不改变原数组元素的顺序.

```
程序名称:C05_010.lua
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30}
for i=1,15 do
    if aa[i]%2==0 then aa[i]=0 end
end
for i=1,15 do io.write(aa[i],",") end
结果为:1,3,0,7,0,0,13,0,0,17,19,21,23,29,0,
```

### §5.3 去掉数组中的某些元素

已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的偶数全部去掉, 得到新的数组:

1, 3, 7, 13, 17, 19, 21, 23, 29

要求不改变原数组元素的顺序.

```
程序名称:C05_020.lua
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30};
for i=15,1,-1 do
    if aa[i]%2==0 then table.remove(aa,i) end
end
for i=1,#aa do io.write(aa[i],",") end
```

该方法由于要将后续的数字向前移动, 所以速度并不快, 下面的方法是高效的. 当 i 走到尾时, j 就是有用数据的最后位置.

```
程序名称:C05_030.lua
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30}
j=0 --j新位置,表示第j位置已经排好
for i=1,#aa do
    if aa[i]%2~=0 then
        j=j+1; aa[j]=aa[i]
    end
end -- j位置之后的数据需要删除
for i=#aa,j+1,-1 do table.remove(aa,i) end
print(table.concat(aa,","))
```

相当于重新构造一个数组

也可以: a[i]=nil 来删除数据

### §5.4 数组元素查询

已知数组具有从小到大的顺序关系:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

键盘输入一个数 x, 判断其是否在数组中.

```
程序名称:C05_040.lua
x=io.read("*number")
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30};
k=0
for i=1,15 do
    if aa[i]==x then k=i; break end
end
if k>0 then
    print("Yes")
else
    print("No")
end
输入:8
输出:Yes
输入:9
输出:No
```

一般满足条件时, 令 k=1

另外, 创建一个标志表, 通过查询标志是否存在, 来实现查询.

```
程序名称:C05_050.lua
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30};
bb={}
for i=1,#aa do bb[aa[i]]=true end
x=io.read("*number")
if bb[x] then
    print("Yes")
else
    print("No")
end
```

这种方法也很巧妙, 先设定 bb[aa[i]]=true, 最后做判断时就直接写 bb[x]

### §5.5 数组元素插入

已知数组具有从小到大的顺序关系:

可以这样写

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19

键盘输入一个数  $x$ , 将其插入到数组中, 并且保持原有顺序.

```

程序名称: C05_060.lua
aa={1,3,4,7,8,10,13,14,16,17,19};
x=io.read("*number")
k=0
for i=#aa,1,-1 do
    if aa[i]<x then k=i; break end
end
table.insert(aa,k+1,x)
for i=1,#aa do io.write(aa[i],",") end
输入: 2
输出: 1,2,3,4,7,8,10,13,14,16,17,19,
  
```

由于本题已知数据的有序性, 故可以使用二分法查找.

### §5.6 生成菲波纳切数列

已知函数

$$F_1 = 1, F_2 = 1, F_n = F_{n-2} + F_{n-1}, n \geq 3$$

输出数列的前 12 项:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

```

程序名称: C05_070.lua
F={1,1}
for i=3,12 do
    F[i]=F[i-2]+F[i-1]
end
for i=1,#F do io.write(F[i],",") end
结果为: 1,1,2,3,5,8,13,21,34,55,89,144,
  
```

### §5.7 分离十进制整数 $a$ 的每一位数字

比如: 输入 45678 系统输出 4,5,6,7,8,

```

程序名称: C05_080.lua
x=io.read("*number")
F={}
for i=1,15 do
    F[i]=x%10
  
```

```

x=(x-F[i])/10
if x==0 then break end
end
for i=#F,1,-1 do io.write(F[i],",") end
  
```

不需要, 新代码如下

另一种实现方案.

```

程序名称: C05_090.lua
x=12345
ss=toString(x)
F={ss:byte(1,-1)}
for i=1,#F do
    io.write(string.char(F[i]),",")
end
  
```

### §5.8 成绩均值与方差

已知学生成绩存于数组中:

81, 83, 84, 77, 78, 80, 93, 94, 88, 60

编写程序, 求其成绩均值、方差.

```

程序名称: C05_100.lua
aa={81,83,84,77,78,80,93,94,88,60}
sum=0
for i=1,10 do sum=sum+aa[i] end
ave=sum/10
dev=0
for i=1,10 do dev=dev+(aa[i]-ave)^2 end
dev=dev/10
io.write("ave=",ave,", dev=",dev)
结果为: ave=81.8, dev=83.56
  
```

### §5.9 集合的交集

已知 A 集合: 81, 83, 83, 83, 84, 80, 93, 94, 88, 60

已知 B 集合: 21, 56, 83, 81, 43, 78, 64, 83, 83

求集合的交集. 注意: 原集合无序且元素有重复, 而输出的交集则要求有序且无重复元素.

```

程序名称: C05_110.lua
A={81,83,83,83,84,80,93,94,88,60}
B={21,56,83,81,43,78,64,83,83}
table.sort(A)
table.sort(B)
C={}
i,j=1,1
while i<=#A and j<=#B do
    if A[i]==B[j] then
  
```

求交集采用了先排序，然后逐序对比的方法，需注意不重复

不重复  
逐序

```
if #C==0 then C[1]=A[i] end
if C[#C]~=A[i] then C[#C+1]=A[i] end
i,j=i+1,j+1
elseif A[i]<B[j] then
i=i+1
else
j=j+1
end
end
for i=1,#C do io.write(C[i],",") end
结果为:81,83,
```

还可以这样写

### §5.10 集合的并集

已知 A 集合:81,83,83,83,84,80,93,94,88,60

已知 B 集合:21,56,83,81,43,78,64,83,83

求集合的并集.

程序名称:C05\_120.lua

A={81,83,83,83,84,80,93,88,60}

B={21,56,83,81,43,78,64,83,83}

table.sort(A)

table.sort(B)

C={}

i,j=1,1

while i<=#A and j<=#B do

if A[i]<=B[j] then

if #C==0 then C[1]=A[i] end

if C[#C]~=A[i] then C[#C+1]=A[i] end

i=i+1;

else

if #C==0 then C[1]=B[j] end

if C[#C]~=B[j] then C[#C+1]=B[j] end

j=j+1

end

end

while i<=#A do

if #C==0 then C[1]=A[i] end

if C[#C]~=A[i] then C[#C+1]=A[i] end

i=i+1;

end

while j<=#B do

if #C==0 then C[1]=B[j] end

if C[#C]~=B[j] then C[#C+1]=B[j] end

j=j+1

end

for i=1,#C do io.write(C[i],",") end

结果为:21,43,56,60,64,78,80,81,83,84,88,93,

求并集同样采用了先排序后逐序对比的方法，一个循环后，最大元素较小的集合已全部遍历，较大的集合将在下一轮循环中从第一轮的位置继续遍历。

第一轮遍历

第二轮遍历

### §5.11 集合的差集 (或补集)

已知 A 集合:81,83,83,83,84,80,93,94,88,60

已知 B 集合:21,56,83,81,43,78,64,83,83

求集合 A-B, 即输出 A 集合中不含 B 集合的元素.

程序名称:C05\_130.lua

A={81,83,83,83,84,80,93,94,88,60}

B={21,56,83,81,43,78,64,83,83}

table.sort(A)

table.sort(B)

C={} ; i=1; j=1

while i<=#A and j<=#B do

if A[i]<B[j] then

if #C==0 then C[1]=A[i] end

if C[#C]~=A[i] then C[#C+1]=A[i] end

i=i+1;

elseif A[i]==B[j] then

i=i+1

else

j=j+1

end

end

while i<=#A do

if #C==0 then C[1]=A[i] end

if C[#C]~=A[i] then C[#C+1]=A[i] end

i=i+1;

end

for i=1,#C do io.write(C[i],",") end

结果为:60,80,84,88,93,94,

对于补集, 即 A 是全集,B 是 A 的子集, 然后求 A-B, 显然上面的程序也适用于求集合的补集.

思路类似求并集

## §5.12 习题

## 1. 已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的首元素移动到数组的尾部, 得到新的数组:

3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30, 1

## 2. 已知数组:

1, 3, 4, 7, 8, 10, 13, 14, 16, 17, 19, 21, 23, 29, 30

编写程序, 将上述数组中的元素倒序, 得到新的数组:

30, 29, 23, 21, 19, 17, 16, 14, 13, 10, 8, 7, 4, 3, 1

## 3. 生成卢卡斯序列:

$$\begin{cases} F_1 = 1, & F_2 = 2, & F_3 = 3 \\ F_n = F_{n-3} + F_{n-2} + F_{n-1}, & n \geq 4 \end{cases}$$

输出数列的前 12 项.

## 4. 用筛法求 100 以内的全部素数

提示: 首先定义一个数组  $aa = \{\}$ , 然后用循环赋初值  $aa[i] = i$ :

1, 2, 3, 4, 5, ..., 100

下面开始划去非素数, 方法是:

二重循环, 外层  $i: 2 \rightarrow 10$  依次检查  $aa[i]$  是否为 0, 若是, 则继续检查下一个, 否则循环去掉  $aa[i]$  的倍数, 去掉的方法是将该值赋值为 0 即可.

原序列:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...

第一次处理(去掉2的倍数):

1, 2, 3, 0, 5, 0, 7, 0, 9, 0, 11, 0, 13, 0, 15, 0, ...

第二次处理(去掉3的倍数):

1, 2, 3, 0, 5, 0, 7, 0, 0, 0, 11, 0, 13, 0, 0, 0, ...

第二次处理(去掉5的倍数):

1, 2, 3, 0, 5, 0, 7, 0, 0, 0, 11, 0, 13, 0, 0, 0, ...

... ..

最后输出大于 1 的非零整数即为所求.

## 5. 杨辉三角

要求产生如下的图形:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1

```

要求产生10行.

提示: 定义一个一维数组  $aa = \{1\}$ , 执行如下的操作即可:

设当前状态为:

```

1 4 6 4 1
从后向前, 将前项加到后项上, 如
1 4 6 4 1-->1
1 4 6 4-->5 1
1 4 6-->10 5 1
1 4-->10 10 5 1
1-->5 10 10 5 1

```

输出结果:

```

1 5 10 10 5 1

```

## 第六章 函数调用

### §6.1 函数定义

系统定义了很多函数,如 sin,cos 等,但有时我们需要自己来定义函数.将具有一定功能的程序块组织在一起,然后给它一个名称,即所谓的函数.函数可能具有多个参数,我们称之为参数列表,参数之间用逗号分隔.例如最大公约数函数.

函数定义的格式为:

```
function 函数名称(参数列表)
    函数体
end
或
函数名称 = function (参数列表)
    函数体
end
```

### §6.2 最大公约数函数

将程序以一个文件的形式存在磁盘上,在需要它的时候直接调用即可.

```
程序名称:MaxGongYueShu.lua
function MaxGongYueShu(x,y)
    if x<0 then x=-x end
    if y<0 then y=-y end
    while x>0 and y>0 do
        if (x>y) then x=x%y else y=y%x end
    end
    return x+y
end
```

由于最大公约数函数是自己定义的,所以使用时需要加一句话来说明它.

```
程序名称:C06_010.lua
require("MaxGongYueShu") -- 声明自定义函数
m=72; n=48
k=MaxGongYueShu(m,n) -- 使用自定义函数
print(k)
结果为:24
```

另外,也可以将自己定义的函数放在主程序之前,这样一个程序文件就包含了全部程序,阅读起来较为方便.通常情况下对于经常使用的函数存在外部文件中,对于使用频率较少的函数定义在主程序的上部.

### §6.3 最小质因数函数

编写一个非常容易理解的简单函数,存放在主程序之上,然后调用它.

```
程序名称:C06_020.lua
function MinZhiYinShu(x) --最小质因数函数
    for i=2,x-1 do
        if (x%i==0) then return i end
    end
    return x
end
--主程序
x=io.read("*number")
k=MinZhiYinShu(x)
print(k)
输入:21
输出:3
```

函数程序放在外部文件时,需用require(" ")函数进行调用,括号内为程序文件名,若函数程序与主程序在一个文件内,则不需要调用函数

这个程序的效率较低,可以编写一个相对复杂,但效率较高的程序存在外部磁盘上,以便于日后调用.

```
程序名称:MinZhiYinShu.lua
function MinZhiYinShu(x)
    if x%2==0 then return 2 end
    for i=3,math.sqrt(x),2 do
        if (x%i==0) then return i end
    end
    return x
end
```

某一个合数的最小因数不超过该数的开方根.

质数直接输出

### §6.4 正整数的质因数分解

利用上面的最小质因数函数可以进行正整数的质因数分解.

比如:

输入 11 输出 11=11  
输入 12 输出 12=2\*2\*3

```
程序名称:C06_030.lua
require("MinZhiYinShu")
m=io.read("*number")
io.write(m,"=")
n=m
while n>1 do
    k=MinZhiYinShu(n)
    if (m>n) then io.write("*") end
    io.write(k)
    n=n/k
end
```

引入·n只是为了判断是否需要输出\*

输入:60

输出:60=2\*2\*3\*5

## §6.5 随机数发生器函数

很多事情都需要用到随机数发生器,尤其是均匀随机数发生器,本文定义了一个  $[0,1)$  区间上的均匀随机数发生器 Rand()。

可以用随机数估计定积分的大小、生成随机试卷等。

设  $y = f(x) = x^2$ , 计算定积分:

$$\int_0^1 x^2 dx$$

频率法

方法是在边长为 1 的正方形内均匀投点, 该定积分就是落在  $y = f(x) = x^2$  曲线下的阴影面积, 也就是落在阴影部分的点数占总点数的比例。

程序名称:Rand.lua

--均匀随机数发生器,周期为2147483648,即21亿

xx={1,0}

function Rand()

xx[2]=xx[2]\*45221+xx[1]\*4793

xx[1]=xx[1]\*45221+453806245

xx[2]=xx[2]+(xx[1]-xx[1]%65536)/65536

xx[2]=xx[2]%32768; xx[1]=xx[1]%65536

return (xx[2]\*65536+xx[1])/2147483648

end

程序名称:C06\_040.lua

require("Rand")

k=0

for i=1,20000 do

x=Rand(); y=Rand()

if x\*x>y then k=k+1 end

end

print(k/20000)

结果为:0.33845

理论为:1/3=0.33333

## §6.6 组合函数

比较难,好好琢磨

输出从 1,2,3,4,5 五个数中任意选取三个数的全部组合。

全部组合如下:

1 2 3

1 2 4

1 2 5

1 3 4

1 3 5

1 4 5

2 3 4

2 3 5

2 4 5

3 4 5

上面的情况很有规律,可以编写一个函数,从一个状态生成下一个状态。首先从后向前,找到第一个能够增加的位置,若不存在增加的位置,则说明已经全部结束了,若找到能够增加的位置,则增加 1,然后在增加的位置后,数字按照升序排列。

程序名称:C06\_050.lua

x={}

function Combination(n,m)

k=0

for i=m,1,-1 do

if x[i]+m-i<n then k=i; break end

end

if k==0 then return 1 end -- over

x[k]=x[k]+1 -- 调本位

for j=k+1,m do x[j]=x[j-1]+1 end

return 0 -- ok!

end

n=5; m=3; --C\_n^m 总个数,组合数

for i=1,n do x[i]=i end -- 初始化

repeat

for i=1,m do io.write(" ",x[i]) end

io.write("\n")

until Combination(n,m)==1

真正输出的是最后一行函数内容

## §6.7 计算今天星期几

已知 2012 年 1 月 1 日是星期日,编写一个函数,输入今天的日期,输出星期几。要求输入三个数,年,月,日,然后输出一个数 0-6,其中 0: 星期日,1: 星期一,...6: 星期六。

程序名称:C06\_060.lua

function RunNian(yy) --是否闰年

if yy%400==0 then return 1 end

if yy%100==0 then return 0 end

if yy%4==0 then return 1 end

return 0

end

日期与时间的互换

先调整第三位,第三位无法调后,调整第二位,同理调整第一位,每次函数映射后,形成一个新的组合。

调前位



```

function YueFen(m) --该月天数
    if m==4 or m==6 or m==9 or m==11 then
        return 30
    end
    if m==2 then return 28 end
    return 31
end

--计算今天是星期几
function XingQi(yy,mm,dd)
    nSum=0
    for nY=2012,yy-1 do
        nSum = nSum + 365 + RunNian(nY)
    end
    for nM=1,mm-1 do
        nSum = nSum + YueFen(nM)
        if nM==2 then nSum=nSum+RunNian(yy) end
    end
    nSum = nSum + dd
    return (nSum-1)%7 --从零开始
end

y,m,d=io.read("*number","*number","*number")
print(XingQi(y,m,d))
xq={"日","一","二","三","四","五","六"}
print("星期"..xq[XingQi(y,m,d)+1])
输入:2016 8 23
输出:2
输出:星期二

```

## §6.8 字符串数组的全连接

尤其是数组内含数组的情况

```

程序名称:C06_070.lua
function concats(tt,ch)
    if type(tt) ~= "table" then return tt end
    local res={}
    for i=1,#tt do
        res[i]=concats(tt[i],ch)
    end
    return table.concat(res,ch or "")
end

aa={
    {"aaa","bbb","ccc"},
    {"eee","ff","gg"},
    {"1","2","345"}
}
print(concats(aa," "))
结果为:aaa bbb ccc eee ff gg 1 2 345

```

参数ch只用于元素间的分隔,可以不要

## §6.9 复杂计数器

重点

某事务有三种属性 每种属性有多种状态, 分别为 (2,4,3), 如何输出该事务的所有状态?

程序名称:C06\_080.lua

aa={} --定义状态空间,每种状态均从1开始  
bb={2,4,3} --定义属性的状态数

```

function NestNum() --产生下一种状态
    aa[#bb]=aa[#bb]+1 --末位置自增
    for i=#bb,2,-1 do --调整
        if aa[i]<=bb[i] then break end
        aa[i-1]=aa[i-1]+1; aa[i]=1 --进位并置1
    end
    if aa[1]>bb[1] then return 1 end --终止态
    return 0 --存在下一种状态
end

```

起始为1, 末数为最大状态数,也是一种组合问题

for i=1,#bb do aa[i]=1 end --初值初始化为0  
k=1 --计数器, 每行输出3个

```

repeat
    for i=1,#bb do io.write(aa[i], " ") end
    io.write(k%3==0 and "\n" or " ")
    k=k+1
until NestNum()==1 --全1状态时退出

```

判断句, 若k%3==0, 则换行, 否则空格

```

1 1 1    1 1 2    1 1 3
1 2 1    1 2 2    1 2 3
1 3 1    1 3 2    1 3 3
1 4 1    1 4 2    1 4 3
2 1 1    2 1 2    2 1 3
2 2 1    2 2 2    2 2 3
2 3 1    2 3 2    2 3 3
2 4 1    2 4 2    2 4 3

```

末尾增优先。后一位增至最大后, 前一位加1, 后一位归零。

如果需要屏蔽某些属性, 比如屏蔽第二位, 即始终保持第二位的某种状态, 可以在添加一个状态数组来实现。

程序名称:C06\_090.lua

```

aa={} --状态空间,每种状态均从1开始
bb={2,4,3} --属性的状态数
cc={0,3,0} --属性标志,0:允许变更
--属性标志的非0值即为该属性的强制状态
function NestNum() --产生下一种状态
    local flag=true
    for i=#cc,1,-1 do --末位置自增
        if cc[i]==0 then
            aa[i]=aa[i]+1; flag=false; break
        end
    end
    if flag then return 1 end

```

这一节先跳过

注意, 此时的末位置指从最后算第一个c[i]为0的位置, 但本质上和上面的情况差不多

检测并调整

```

for i=#bb,1,-1 do --全检测并全调整
    if aa[i]>bb[i] then
        pp=0
        for j=i-1,1,-1 do
            if cc[j]==0 then pp=j; break end
        end
        if pp~=0 then
            aa[pp]=aa[pp]+1; aa[i]=1 --进位并置1
        else return 1 end --终止态
        end
    end
    return 0 --存在下一种状态
end

for i=1,#bb do
    if cc[i]==0 then
        aa[i]=1 --初值初始化为1
    else
        aa[i]=cc[i] --强制属性不变更
    end
end
repeat
    for i=1,#bb do io.write(aa[i]," ") end
    io.write("\n")
until NestNum()==1 --全1状态时退出
结果为:
1 3 1
1 3 2
1 3 3
2 3 1
2 3 2
2 3 3

```

## §6.10 习题

1. 输出 100 以内的所有正整数的质因数分解.

```

1=1
2=2
3=3
4=2*2
5=5
6=2*3
7=7
8=2*2*2
9=3*3
...
96=2*2*2*2*2*3
97=97
98=2*7*7
99=3*3*11
100=2*2*5*5

```

2. 编写一个插入排序的函数.

全局变量为整型数组 A 及其元素个数 n, 函数的入口参数为待插入元素 x, 假设 n 小于数组的大小.

编写一个主程序, 调用上述子程序, 数据为 1,3,4,7,8,10,13,14,16; 键盘输入待插入数值.

3. 编写一个针对整型数组的二分法的查找函数.

全局变量为整型数组 A 及其元素个数 n, 函数的入口参数为待查询元素 x.

编写一个主程序, 调用上述子程序, 数据为 1,3,3,3,7,7,13,14,15,17,19,21,23,29,30. 键盘输入待查数据, 然后显示“在数组中”或“不在数组中”.

4. 编写产生杨辉三角的函数.

全局变量为整型数组 A, 数组的总大小为 n, 假设当前行状态已知, 编写函数生成下一行状态.

编写主程序调用上述子程序, 要求产生 10 行.

5. 打印出从 2002 年到 2079 年间的全部闰年的个数.

## 第七章 二维数组

### §7.1 二维数组的定义及初始化

ab={{1,5,3},{6,8,9}}

### §7.2 矩阵加法

已知矩阵 A,B, 求 A+B.

$$A = \begin{bmatrix} 1 & 5 & 3 & 3 \\ 4 & 2 & 1 & 3 \\ 3 & 2 & 0 & 3 \end{bmatrix}, B = \begin{bmatrix} 2 & 1 & 0 & 3 \\ 2 & 3 & 1 & 3 \\ 3 & 1 & 2 & 3 \end{bmatrix}$$

程序名称:C07\_010.lua

A={{1,5,3,3},{4,2,1,3},{3,2,0,3}}

B={{2,1,0,3},{2,3,1,3},{3,1,2,3}}

C={}

for i=1,#A do

C[i]={}

for j=1,#A[i] do

C[i][j]=A[i][j]+B[i][j]

end

end

for i=1,#A do

io.write(table.concat(C[i],"\t"),"\n")

end

结果为:

3 6 3 6  
6 5 2 6  
6 3 2 6

定义列组时先定义行组

列数

串联同一行字符,用空格隔开

### §7.3 矩阵的消法变换

已知矩阵:

$$A = \begin{bmatrix} 1 & 5 & 3 & 3 \\ 4 & 2 & 1 & 3 \\ 3 & 2 & 0 & 3 \end{bmatrix}$$

将第3行乘以4,加到第1行上去.

程序名称:C07\_020.lua

A={{1,5,3,3},{4,2,1,3},{3,2,0,3}}

for j=1,#A[1] do

A[1][j] = A[1][j] + 4\*A[3][j]

end

for i=1,#A do

io.write(table.concat(A[i],"\t"),"\n")

end

结果为:

13 13 3 15  
4 2 1 3  
3 2 0 3

### §7.4 矩阵的转置变换

已知矩阵:

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix}$$

方阵转置的本质在于非对角元的对换

将矩阵 A 转置,即将 i,j 位置的元素互换.

程序名称:C07\_030.lua

A={{1,5,3},{4,2,1},{3,2,0}}

for i=1,#A do

for j=1,i-1 do

A[i][j],A[j][i]=A[j][i],A[i][j]

end

end

for i=1,#A do

io.write(table.concat(A[i],"\t"),"\n")

end

结果为:

1 4 3  
5 2 2  
3 1 0

注意第二个循环语句中应该写i-1而不是#A[i]

对于行列不相同的情况,最好重新开辟空间,如果还放在原来的矩阵中则可能会出现过多的元素,去掉这些多出来的元素依然是一件麻烦事.

已知矩阵:

$$A = \begin{bmatrix} 1 & 5 & 3 & 5 & 3 \\ 4 & 2 & 1 & 5 & 3 \\ 3 & 2 & 0 & 5 & 3 \end{bmatrix}$$

程序名称:C07\_040.lua

A={{1,5,3,5,3},{4,2,1,5,3},{3,2,0,5,3}}

B={}

for j=1,#A[1] do

B[j]={}

for i=1,#A do

B[j][i] = A[i][j]

end

end

for i=1,#B do

io.write(table.concat(B[i],"\t"),"\n")

end

结果为:

1 4 3  
5 2 2  
3 1 0  
5 5 5  
3 3 3

没有必要重新建一个矩阵,元素对换中非空元和空元也能对换

行数

## §7.5 学生成绩

现有数学、语文、外语、政治、物理、化学六门课程成绩如下:

$$A = \begin{bmatrix} 81 & 75 & 83 & 81 & 75 & 83 \\ 44 & 92 & 71 & 53 & 82 & 80 \\ 93 & 52 & 80 & 54 & 92 & 71 \end{bmatrix}$$

1. 求理科成绩平均分 (数学、物理、化学).
2. 统计不及格的人数和门数.

程序名称:C07\_050.lua

--理科平均分

aa={

{81, 75, 83, 81, 75, 83},

{44, 92, 71, 53, 82, 80},

{93, 52, 80, 54, 92, 71}

}

s=0

k=0

for i=1,3,1 do

s=s+aa[i][1] + aa[i][5] + aa[i][6]

k=k+3

end

io.write("平均分"..s/k.."n")

--不及格人数

rs=0

for i=1,3,1 do

for j=1,6,1 do

if aa[i][j]<60 then

rs=rs+1

break

end

end

end

io.write("人数"..rs.."n")

--不及格门数

ms=0

for j=1,6,1 do

for i=1,3,1 do

if aa[i][j]<60 then

ms=ms+1

break

end

end

end

io.write("门数"..ms.."n")

## §7.6 习题

1. 矩阵乘法, 求 AB:

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 & 0 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

2. 矩阵的换法变换

交换矩阵的第1行和第3行, 其中矩阵如下:

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix}$$

3. 计算矩阵的所有元素之和

$$A = \begin{bmatrix} 1 & 5 & 3 \\ 4 & 2 & 1 \\ 3 & 2 & 0 \end{bmatrix}$$

4. 用二维数组存储杨辉三角, 然后打印.

要求产生如下的图形:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

要求产生10行.

5. 用二维数组存储图形, 然后打印.

图形如下:(三对角矩阵)

```
1 1 0 0 0 0 0
1 2 1 0 0 0 0
0 3 3 1 0 0 0
0 0 5 4 1 0 0
0 0 0 7 5 1 0
0 0 0 0 9 6 1
0 0 0 0 0 11 7
```

6. 二维数组存储图形并打印.(蛇行三角)

```
22 0 0 0 0 0 0
16 23 0 0 0 0 0
11 17 24 0 0 0 0
7 12 18 25 0 0 0
4 8 13 19 26 0 0
2 5 9 14 20 27 0
1 3 6 10 15 21 28
```

## 第八章 堆栈技术

利用一维数组来构造一种数据存储形式，通过形象的动作序列来表现数据的处理过程。

### §8.1 堆栈的一般形式

```
ss={}; --堆栈定义
ss[#ss+1]=k; --将k压栈
k=ss[#ss]; ss[#ss]=nil --出栈,到k中
```

### §8.2 十进制转换为二进制

例如: 42 --> 101010  
27 --> 11011

```
程序名称:C08_010.lua
ss={}
k=io.read("*number")
while k>0 do
    ss[#ss+1] = k%2 --变量k的结果压栈
    k=(k-k%2)/2
end
while #ss>0 do
    io.write(ss[#ss]); ss[#ss]=nil
end
```

入栈

出栈

二进制转换为十进制程序。

```
程序名称:C08_020.lua
k=0
ss=io.read("*line")
for i=1,#ss do
    k=k*2
    if ss:sub(i,i)=="1" then k=k+1 end
end
print(k)
```

读取字符串

方法很巧妙！

### §8.3 回文数

一个 5 位数，判断它是不是回文数，即 12321 是回文数，个位与万位相同，十位与千位相同。

```
程序名称:C08_030.lua
ss={}
k=io.read("*number")
while k>0 do
    ss[#ss+1]=k%10
    k=(k-k%10)/10
end
print(table.concat(ss,""))
```

对称数

也可以先读取数字的字符串，提取每一位放入数组中

```
if ss[1]==ss[5] and ss[2]==ss[4] then
    print("Yes")
else
    print("No")
end
```

### §8.4 质因数分解

例如:  $100=2*2*5*5$

程序名称:C08\_040.lua

```
ss={}
k=io.read("*number")
n=k
while n>1 do
    m=n
    for i=2,n-1 do
        if n%i==0 then
            m=i
            break
        end
    end
    ss[#ss+1]=m
    n=(n-n% m)/m
end
print(k.."="..table.concat(ss,""))
```

$n \% m = 0 \dots$

### §8.5 数组数据翻转

设一个有 10 个元素的数组，现将前后数据翻转。这里想介绍一个常规方法，利用两个指针，从前向后和从后向前，同时向中间推进，在行进过程中交换对应数据，直到两个指针相遇时终止。

原数组: 1,2,3,4,5,9,8,7,6,5  
翻转后: 5,6,7,8,9,5,4,3,2,1

```
程序名称:C08_050.lua
aa={1,2,3,4,5,9,8,7,6,5};
p1=1; p2=#aa
while p1<p2 do
    aa[p1],aa[p2]=aa[p2],aa[p1]
    p1=p1+1; p2=p2-1
end
io.write(table.concat(aa,""))
```

妙极了

### §8.6 字符串反序

利用堆栈缓存，然后倒序输出。

程序名称:C08\_060.lua

ss=io.read("\*line")

aa={}

for i=1,#ss do aa[i]=ss:sub(-i,-i) end

pp=table.concat(aa)

print(pp)

倒序提取字符，  
即从#ss开始

## §8.7 复杂连分数计算

设复杂连分数:

$$\frac{1}{2 + \frac{3}{4 + \frac{5}{6}}} \quad \text{记为: } \frac{1}{2} + \frac{3}{4} + \frac{5}{6}$$

一般连分数记为:

$$\frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_n}{b_n}$$

编程求解:

$$\frac{1}{b_1} + \frac{1}{b_2} + \cdots + \frac{1}{b_n}$$

其中  $b_1, b_2, \dots, b_n$  均为非零实数, 要求最后增加输入一个 0, 它不参与计算, 仅表示输入结束。

例如: 输入 1 2 3 0 输出 0.7

程序名称:C08\_070.lua

ss={}

while true do

k=io.read("\*number")

if k==0 then break end

ss[#ss+1]=k --变量k的结果压栈

end

jg=0

for i=#ss,1,-1 do

jg=1/(ss[i]+jg)

end

print(jg)

连分数的应用非常广泛, 任何实数都可以使用连分数逼近, 如:

$$\frac{\sqrt{5}-1}{2} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \cdots$$

$$\sqrt{2}-1 = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \cdots$$

## §8.8 习题

### 1. 十进制转换为十六进制

例如: 输入 123 输出 7B

输入 98765 输出 181CD

输入 12345789 输出 75BCD15

### 2. 数位分离

一个  $2n$  位数, 将其分离成奇数位在前, 偶数位在后的一个新数。

例如: 123456 --> 135246

36235789 --> 32586379

### 3. 阶乘的质因数分解

求  $n!$ , 其中  $10 < n < 50$ , 要求相同质因数合并用  $\wedge$  来表示, 例如:

例如:  $13! = 2^{10} * 3^5 * 5^2 * 7^1 * 11^1 * 13^1$

注意: 不要将  $n!$  计算出来, 否则可能会超出整型数值的表示范围。

提示: 显然, 13 以内的所有素数都会在分解中出现, 而且 13 中含有 2 的因子个数为

$$[13/2] + [13/4] + [13/8] + [13/16] + [13/32] + \cdots$$

$$= 6 + 3 + 1 + 0 + 0 + \cdots = 10$$

## 第九章 函数递归

函数内直接或间接调用自己本身,称之为递归调用. 思维简单编程快.

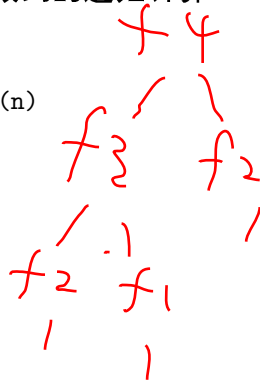
### §9.1 十进制转换为二进制

例如: 42 --> 101010  
27 --> 11011  
程序名称: C09\_010.lua  
function JZ(n)  
if n<=1 then  
io.write(n)  
else  
JZ((n-n%2)/2)  
io.write(n%2)  
end  
end  
k=io.read("\*number")  
JZ(k)

千万注意, 多层嵌套函数递归的思想是先执行, 然后逐步退回上一层执行后续操作, 如左: 当n=3时, 先执行第二层JZ((3-3%2)/3), 输出1, 再退回第一层JZ, 执行未剩下的语句, 输出第二个1, 即3%2

### §9.2 菲波纳切数列的递归计算

数列: 1, 1, 2, 3, 5, 8, ...  
公式:  $F(n+2)=F(n+1)+F(n)$   
求F(12).  
程序名称: C09\_020.lua  
function F(n)  
if n<=2 then  
return 1  
end  
return F(n-1)+F(n-2)  
end  
print(F(12))



经典例题

### §9.3 递归计算杨辉三角

计算杨辉三角的第5行.  
公式:  $F(m, 1)=1$   
公式:  $F(m, n)=0, n>m$   
公式:  $F(m, n)=F(m-1, n-1)+F(m-1, n)$   
数列: 1 5 10 10 5 1  
程序名称: C09\_030.lua  
function YH(m, n)  
if n==0 then return 1 end  
if m<=n then return 0 end  
return (YH(m-1, n-1)+YH(m-1, n))  
end  
for i=0, 5 do

```
io.write(YH(6,i),",")  
end
```

### §9.4 伞型序递归

一个正整数n, 不用循环和其他变量, 按照n, 2n, 4n, 8n, ... 的顺序递增, 当值大于5000时, 把值按照原序和反序输出来.

例: n=1237, 则输出为:

参考答案有误

1237, 2474, 4948, 9896, 9896, 4948, 2474, 1237,

程序名称: C09\_040.lua  
function Out(m)  
io.write(m..",")  
if m<5000 then  
Out(2\*m)  
io.write(m..",")  
end  
end  
n=io.read("\*number")  
Out(n)

```
function out(n)  
io.write(n,"\t")  
if n<=5000 then  
out(2*n)  
end  
io.write(n,"\t")  
end  
n=io.read("*number")  
out(n)
```

### §9.5 利用递归求最大公因子

程序名称: C09\_050.lua  
function gyz(a, b)  
if a==0 or b==0 then  
return(a+b)  
end  
if a>b then  
return(gyz(a%b, b))  
else  
return(gyz(b%a, a))  
end  
end  
print(gyz(72, 48))

```
function gyz(m, n)  
if n==0 or m==0 then  
io.write(m+n)  
elseif m>n then  
gyz(n, (m%n))  
else  
gyz((n%m), m)  
end  
end  
gyz(24, 48)
```

### §9.6 对策局势的确定

比较难理解, 先略过

有一堆火柴15根, 甲乙二人轮流拿, 每次最多拿三根, 最少拿一根, 谁最后拿没谁赢, 设甲先拿, 问对甲而言此局势是赢局还是输局?

程序名称: C09\_060.lua  
function JS(n)



```

if n>=1 and n<=3 then return 1 end --赢
for i=1,3 do
    if JS(n-i)==0 then return 1 end
end
return 0
end
print(JS(15))

```

### §9.7 利用递归实现连分数计算

编程求解:

经典  
例题

$$\frac{1}{b_1} + \frac{1}{b_2} + \cdots + \frac{1}{b_n}$$

其中  $b_1, b_2, \dots, b_n$  均为非零实数, 要求最后增加输入一个 0, 它不参与计算, 仅表示输入结束.

例如: 输入 1 2 3 0 输出 0.700000

```

程序名称:C09_070.lua
function JS()
    k=io.read("*number")
    if k==0 then return 0 end
    return 1/(k+JS())
end
print(JS())

```

### §9.8 递归求根

函数:

$$f(x) = x^3 - 5x + 1, f(0) > 0, f(1) < 0,$$

求  $f(x) = 0$  在  $(0,1)$  上的一个实根.

```

程序名称:C09_080.lua
function gen(a,b)
    c=(a+b)/2
    if b-a<0.001 then return c end
    if c*c*c-5*c+1>0 then return gen(c,b) end
    return gen(a,c)
end
print(gen(0,1))

```

非递归二分法求根.

```

程序名称:C09_090.lua
a=0 b=1
while b-a>0.001 do
    c=(a+b)/2
    if c*c*c-5*c+1>0 then a=c else b=c end
end
print((a+b)/2)

```

### §9.9 习题

#### 1. 用递归实现十进制转换为十六进制

例如: 输入 123 输出 7B

输入 98765 输出 181CD

输入 12345789 输出 75BCD15

#### 2. 求卢卡斯序列

数列: 1, 2, 3, 6, 11, 20, ...

公式:  $F(1)=1, F(2)=2, F(3)=3$

公式:  $F(n+3)=F(n+2)+F(n+1)+F(n)$

求  $F(12)$ .

#### 3. 利用递归方法求 5!

$5!=1*2*3*4*5$

公式:  $F(1)=1, F(n)=n*F(n-1)$

求  $F(5)$ .

4. 有 5 个人坐在一起, 问第五个人多少岁? 他说比第 4 个人大 2 岁。问第 4 个人岁数, 他说比第 3 个人大 2 岁。问第三个人, 又说比第 2 人大两岁。问第 2 个人, 说比第 1 个人大两岁。最后问第 1 个人, 他说是 10 岁。请问第五个人多大?

这也用递归?

5. 给一个不多于 5 位的正整数, 要求: 一、求它是几位数, 二、逆序打印出各位数字。

#### 6. 对策局势的确定

有两堆火柴分别为 7, 10 根, 甲乙二人轮流拿, 要求每次最多拿三根, 最少拿一根, 并且每次只能在其中的一堆中取, 设甲先拿, 问对甲而言此局势是赢局还是输局?

#### 7. 对策局势的确定

有两堆火柴分别为 7, 10 根, 甲乙二人轮流拿, 每次都有两种取法, 第一种: 要求每次最多拿三根, 最少拿一根, 并且只能在其中的一堆中取, 第二种: 在两堆中同时取, 并且每堆取相同数目, 要求每堆最多拿三根, 最少拿一根, 设甲先拿, 问对甲而言此局势是赢局还是输局?

#### 8. 对策局势的确定

有三堆火柴分别为 3, 4, 5 根, 甲乙二人轮流拿, 要求每次最少拿一根, 最多可以将整堆全拿走, 并且每次只能在其中的一堆中取, 设甲先拿, 问对甲而言此局势是赢局还是输局?

关于return  
语句





io.input没啥用，io.output既可以在已经创建的文件中编辑（会覆盖原文件内容），也可以自动创建新文件再编辑。

## 第十章 文件操作

### §10.1 文件操作的定义

重新定义当前输入输出文件：

```
io.input("文件名称")  
io.output("文件名称")  
io.close()
```

打开文件：

```
fp=io.open("文件名称","r")  
fp=io.open("文件名称","w")  
fp=io.open("文件名称","rb")  
fp=io.open("文件名称","wb")
```

可以使用assert命令强制显示出错信息。

关闭文件：

```
fp:close()
```

读文件全部内容：

```
io.read("*all")  
fp:read("*all")
```

写文件：

```
io.write(kkk)  
fp:write(kkk)
```

注意，io.open()貌似只能和assert搭配使用，但兼具了读写两个功能，比io.output强

### §10.2 写文件

将星型图形写入文本文件中。

```
*  
**  
***  
****  
*****  
程序名称:C10_010.lua  
io.output("星型图形.txt","w")  
for i=1,5 do  
    for j=1,i do io.write("*"); end  
    io.write("\n")  
end  
io.close()
```

### §10.3 数组内容写文件

将数组中的数据写入文本文件中。

```
13 5 6 3 7 9 2 11 54 33  
程序名称:C10_020.lua  
aa={13,5,6,3,7,9,2,11,54,33}  
fp=assert(io.open("数列1.txt","w"))  
for i=1,#aa do  
    fp:write(aa[i], " ")  
end  
fp:close()
```

### §10.4 读文件并存于数组中

数据存在文件中，读入文件中的数据并存于数组中。

```
13 5 6 3 7 9 2 11 54 33  
程序名称:C10_030.lua  
aa={}  
fp=assert(io.open("数列1.txt","r"))  
for i=1,10 do  
    aa[i]=fp:read("*number")  
end  
fp:close()  
for i=1,#aa do  
    io.write(aa[i], " ")  
end
```

### §10.5 将二维数组内容写入文件中

数据存在于二维数组中，将数据写入到外部文件中。

```
12 13 14 15 16  
22 23 24 25 26  
32 33 34 35 36  
程序名称:C10_040.lua  
aa={}  
fp=assert(io.open("数列2.txt","w"))  
for i=1,3 do  
    aa[i]={}  
    for j=1,5 do  
        aa[i][j]=10*i+j+1  
    end  
end  
for i=1,3 do  
    for j=1,5 do  
        fp:write(aa[i][j], " ")  
    end  
    fp:write("\n")  
end  
fp:close()
```

### §10.6 从文件中读入数据到二维数组中

数据存在于外部文件中，将数据读入到二维数组中。

```
12 13 14 15 16  
22 23 24 25 26  
32 33 34 35 36
```

```

程序名称:C10_050.lua
aa={}
fp=assert(io.open("数列2.txt","r"))
for i=1,5 do
    aa[i]={}
    for j=1,5 do
        aa[i][j]=fp:read("*number")
    end
end
fp:close()
for i=1,5 do
    for j=1,5 do
        io.write(aa[i][j], " ")
    end
    io.write("\n")
end

```

矩阵的展示不用这种表达

### §10.7 不定长数据的存储

数据存在于二维数组中, 将数据写入到外部文件中. 数据共有 4 行, 其中第一个元素表示该行元素个数.

```

5 1 2 3 4 5
2 12 13
4 22 23 24 25
7 32 33 34 35 3 3 9

```

与矩阵型二维数组读写类似, 只需要注意内循环语句为 `j=1,#a[i]`

```

程序名称:C10_060.lua
aa={{1,2,3,4,5},
    {12,13},
    {22,23,24,25},
    {32,33,34,35,3,3,9}}
fp=assert(io.open("数列3.txt","w"))
for i=1,#aa do
    fp:write("#aa[i], " ")
    for j=1,#aa[i] do
        fp:write(aa[i][j], " ")
    end
    fp:write("\n")
end
fp:close()

```

### §10.8 不定长数据的读取

数据存在于外部文件中, 将数据读入到二维数组中.

```

1 2 3 4 5
12 13
22 23 24 25
32 33 34 35 3 3 9

```

```

程序名称:C10_070.lua
fp=assert(io.open("数列3.txt","r"))
aa={}
for i=1,4 do
    aa[i]={}
    k=fp:read("*number")
    for j=1,k do
        aa[i][j]=fp:read("*number")
    end
end
fp:close()
for i=1,4 do
    for j=1,#aa[i] do
        io.write(aa[i][j], " ")
    end
    io.write("\n")
end

```

### §10.9 文件合并

先以写方式打开两个文件“文本 1.txt”和“文本 2.txt”, 各存放一行文字:“成小事靠才能,”和“成大事靠品德.”, 关闭这两个文件, 再以读方式打开这两个文件, 并把这两个文件中的信息合并, 输出到一个新文件“文本 3.txt”中.

```

程序名称:C10_080.lua
fp1=assert(io.open("文本1.txt","w"))
fp1:write("成小事靠才能,")
fp1:close()
fp2=assert(io.open("文本2.txt","w"))
fp2:write("成大事靠品德.")
fp2:close()

```

```

fp1=assert(io.open("文本1.txt","rb"))
fp2=assert(io.open("文本2.txt","rb"))
fp3=assert(io.open("文本3.txt","wb"))

```

```

while true do
    ss=fp1:read("*line")
    if not ss then break end
    fp3:write(ss)
end

```

将一个文件的内容复制到另一个文件, 直到读取内容为空

```

while true do
    ss=fp2:read("*line")
    if not ss then break end
    fp3:write(ss)
end
fp1:close()
fp2:close()
fp3:close()

```

## §10.10 文件内容比较

比较两个文件的内容是否完全一模一样. 完全一模一样输出 ok!, 否则输出不一样的附近 16 个字节信息.

程序名称: C10\_090.lua

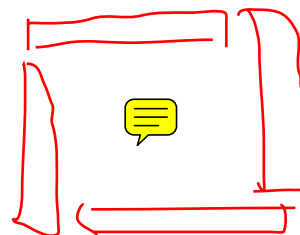
```
fp1=assert(io.open("文本1.txt","rb"))
fp2=assert(io.open("文本2.txt","rb"))
local block=16
local flag=false
local p1,p2,s1,s2
while true do
    s1=fp1:read(block)
    s2=fp2:read(block)
    if not s1 then break end
    if not s2 then break end
    if #s1~=#s2 then
        flag=true
    else
        p1={string.byte(s1,1,-1)}
        p2={string.byte(s2,1,-1)}
        for i=1,block do
            if p1[i]~=p2[i] then
                flag=true; break
            end
        end
    end
end
if flag then
    for _,b in ipairs(p1) do
        io.write(string.format("%02X ",b))
    end
    io.write(string.rep(" ",block-#s1))
    io.write(" ",s1:gsub("%c","."),"\n")
    for _,b in ipairs(p2) do
        io.write(string.format("%02X ",b))
    end
    io.write(string.rep(" ",block-#s2))
    io.write(" ",s2:gsub("%c","."),"\n")
    break
end
end
```

if flag==false then print("ok!") end  
fp1:close(); fp2:close()  
结果为:(因排版原因,省略了中间的部分内容)  
B3 C9 D0 A1 ... DC 2C 成小事靠才能,  
B3 C9 B4 F3 ... C2 2E 成大事靠品德.

## §10.11 打印图形

图形如下: 要求先定义二维数组, 然后添数, 最后将整个数组输出到文件中.

```
1 2 3 4 5 6 7
24 25 26 27 28 29 8
23 40 41 42 43 30 9
22 39 48 49 44 31 10
21 38 47 46 45 32 11
20 37 36 35 34 33 12
19 18 17 16 15 14 13
```



程序名称: C10\_100.lua

```
fp=assert(io.open("数字图形.txt","w"))
N=7
A={};
for i=1,N do A[i]={} end
k=1
for n=1,(N+1)/2 do
    i=n; j=n; A[i][j]=k;
    while j<=N-n do A[i][j]=k; j=j+1; k=k+1 end
    while i<=N-n do A[i][j]=k; i=i+1; k=k+1 end
    while j>n do A[i][j]=k; j=j-1; k=k+1 end
    while i>n do A[i][j]=k; i=i-1; k=k+1 end
end
for i=1,N do
    for j=1,N do
        fp:write(string.format("%4d",A[i][j]))
    end
    fp:write("\n")
end
fp:close()
```

## §10.12 常用命令

tostring(value) -- 数值转换为字符串  
tonumber(str) -- 字符串转换为数值  
x%1 -- 当x为实数时,结果是x的小数部分  
x-x%1 -- 当x为实数时,结果是x的整数部分  
x-x%0.01 -- 当x为实数时,精确到小数点后两位  
a%b -- 结果的符号永远与第二个参数相同  
a..b -- 字符串连接 print(a..b)  
io.lines -- 迭代文件中的每行  
pairs -- 迭代表元素  
ipairs -- 迭代数组元素  
string.gmatch -- 迭代字符串中的单词  
select("#",...) -- 返回变长参数的个数  
select(i,...) -- 返回它的第i个变长参数  
type(表达式) -- 返回结果的类型字符串  
unpack() -- 返回数组所有元素

## §10.13 习题

1. 将星型图形写入文本文件中.

```
*  
***  
*****  
*****  
*****
```

2. 将菲波纳切数列的前 12 项写入到外部文件中.

F1=1, F2=1, F3=2, ...

3. 已知文件中存放了如下数据:

13,5,6,3,7,9,2,11,54,33,

从文件中读入这些数据, 然后计算这些数据的均值. 提示: 均值为 14.3

4. 已知存有不定长数据的文件:

```
5,1,2,3,4,5,  
2,12,13,  
4,22,23,24,25,  
7,32,33,34,35,3,3,9,
```

问, 这个文件中总共存储了多少个有用的数据?  
要求不开辟数组空间.

提示: 不在内存中保留文件中的数据, 累计每一行的第一个元素即可, 答案为 18.

## 第十一章 综合例题

### §11.1 利润提成的计算

例. 企业发放的奖金根据利润提成. 利润(I) 低于或等于 10 万元时, 奖金可提 10%; 利润高于 10 万元, 低于 20 万元时, 低于 10 万元的部分按 10% 提成, 高于 10 万元的部分, 可提成 7.5%; 20 万到 40 万之间时, 高于 20 万元的部分, 可提成 5%; 40 万到 60 万之间时高于 40 万元的部分, 可提成 3%; 60 万到 100 万之间时, 高于 60 万元的部分, 可提成 1.5%, 高于 100 万元时, 超过 100 万元的部分按 1% 提成, 从键盘输入当月利润 I, 求应发放奖金总数? 输入数据以万元为单位, 例如 2.3 代表 2.3 万元.

方法一:(直接使用判断完成计算)

简单分析: 对于一个给定的利润 a, 我们需先判断它属于题目中的哪一种情况, 判断出属于哪一类后就可根据相应的数据计算提成.

程序名称:C11\_010.lua

```
a=io.read("*number")
if a<=10 then
    b=a*0.1
elseif a<=20 then
    b=10*0.1+(a-10)*0.075
elseif a<=40 then
    b=10*0.1+10*0.075+(a-20)*0.05
elseif a<=60 then
    b=10*0.1+10*0.075+20*0.05+(a-40)*0.03
elseif a<=100 then
    b=10*0.1+10*0.075+20*0.05+20*0.03
    b=b+(a-60)*0.015
else
    b=10*0.1+10*0.075+20*0.05+20*0.03
    b=b+40*0.015+(a-100)*0.01
end
print(b)
```

方法二:(使用数组完成计算)

简单分析: 把相应的利润和超出该利润的提成放到一个二维数组 a[] 中, 其中 a[i][1] 为当前利润, a[i][2] 为对应的提成比例, 对于一个给定的利润, 把它和 a[i][1] 进行比较, 判断它的类别, 然后进行计算.

程序名称:C11\_020.lua

```
c={--每组两个数:当前利润,超过此值的提成比例.
    {0,0.1},
```

```
{10,0.075},
{20,0.05},
{40,0.03},
{60,0.015},
{100,0.01},
}

a=io.read("*number")
b=0
for i=1,5 do
    if a<=c[i+1][1] then
        b=b+(a-c[i][1])*c[i][2]
        break
    else
        b=b+(c[i+1][1]-c[i][1])*c[i][2]
    end
end
if a>c[6][1] then
    b=b+(a-c[6][1])*c[6][2]
end
print(b)
```

### §11.2 约瑟夫问题

没看懂

有 n 个人围成一圈, 顺序排号. 从第一个人开始报数 (从 1 到 3 报数), 凡报到 3 的人退出圈子, 问最后留下的人是几号.

程序名称: C11\_030.lua

```
n=10 -- n个人
a={}
for i=1,n do a[i]=i end --初始化

i,k=1,1
while #a>1 do --剩1人时停止
    k=k+1; i=i+1
    if i>#a then i=1 end
    if k==3 then --此人出局
        table.remove(a,i)
        k=1 --重新开始报数
        if i>#a then i=1 end
    end
end
print(a[1]) -- 剩下的号码
```

还有另外一种方法, 即递归算法, 递推公式为:

```
a=0;
a=(a+ k)%i
```

程序名称: C11\_040.lua

```
n=10;k=3 --10个人,报数到3
```

```

a=0 -- 取余运算从0开始
for i=1,n do
    a=(a+k)%i
end
print(a+1) --人的序号从1开始

```

### §11.3 狼找兔子

一只兔子躲进了  $n$  个环形分布的洞的某一个中. 狼在第一个洞没有找到兔子, 就隔一个洞, 到第三个洞去找; 也没有找到, 就隔两个洞, 到第六个洞去找. 以后每次多一个洞去找兔子……这样下去, 如果一直找不到兔子, 请问兔子可能在哪个洞中? (设  $n=19$ )

分析: 从1号洞开始找, 则找到的洞穴序号如下:  $1, 1+2, 1+2+3, \dots$ , 通项公式:  $k(k+1)/2$ , 由环形, 上式需要  $\text{mod } n$ , 故只需要测试从1到  $n$  对应的洞穴号即可.

```

程序名称:C11_050.lua
n=19
a={}
for k=1,n do
    a[(k*(k+1)/2)%n]=1
end
if a[0] then a[n]=a[0] end -- 0号就是n号
for k=1,n do
    if not a[k] then
        print(k)
    end
end

```

### §11.4 包含3的数的个数

例. 从1到10000之间有多少个数含有数字3, 例如131,234等等.

```

程序名称: C11_060.lua
count=0
for i=1,10000 do
    x=i
    while x>0 do
        if x%10==3 then
            count=count+1
            break
        end
        x=(x-x%10)/10
    end
end
print(count)

```

### §11.5 表达式实例

```

程序名称:C11_070.lua
a=8%3; print(a)      --2
a=8.12%3; print(a)   --2.12
a=8.12%1; print(a)   --0.12
a=8.12%0.1; print(a) --0.02
a=8.12%0.01; print(a) --0.009999999999999998
a=8-8%3; print(a)    --6
a=8.12-8.12%3; print(a) --6
a=8.12-8.12%1; print(a) --8
a=8.12-8.12%0.1; print(a) --8.1
a=8.12-8.12%0.01; print(a) --8.11
a=8.12-8.12%0.001; print(a) --8.119
a=8.12-8.12%0.0001 print(a) --8.1199
a={10,20,30}
print(a)              --table: 18BAA798
print(unpack(a))      --10 20 30
a=6
print(a)              --6
print(not a)          --false
print(nil)            --nil
print(not nil)        --true
print(not false)      --true
print(b)              --nil, 因为b变量不存在
print(not b)          --true
print(0)              --0
print(not 0)          --false
print(not not 0)      --true
-- a and b or c 相当于C语言上的 a ? b:c
-- 要求 b 不能为 nil 或 false
m=5; b=3; c=4
print(m<4 and b or c) --4
print(m>4 and b or c) --3
print(10 == 10)       --true
print(10 == "10")     --false
print(10 + 15)        --25
print(10 .. 15)       --1015
print("10" + 15)      --25
print("10" .. 15)     --1015
print("10" * 15)       --150
print("10" / 5)        --2
print("10" - 4)        --6
print("-2.235e-8" * 3) -- -6.705e-008
print() -- 空时,换行
ss="abc def hh"
print(string.sub(ss,5,8)) --def
print(ss:sub(5,8))       --def
print(ss:find(" ",1))   --4 4
print(ss:find(" ",5))   --8 8

```

## 习题参考答案

### 第 1 章答案

程序名称:D01\_01.lua

```
a=math.exp(math.sin(18/180*math.pi))
b=math.log10(42)
c=math.exp(12*math.log(13))
print(a*b+c)
```

程序名称:D01\_02.lua

```
print(11/(12+333/(13+23/5)))
```

程序名称:D01\_03.lua

```
a11=1; a12=1; a13=1; b1=1
a21=2; a22=2; a23=1; b2=3
a31=3; a32=4; a33=5; b3=5

cc0=a11*a22*a33+a12*a23*a31+a13*a21*a32
-a13*a22*a31-a11*a23*a32-a12*a21*a33
cc1=b1*a22*a33+a12*a23*b3+a13*b2*a32
-a13*a22*b3-b1*a23*a32-a12*b2*a33
cc2=a11*b2*a33+b1*a23*a31+a13*a21*b3
-a13*b2*a31-a11*a23*b3-b1*a21*a33
cc3=a11*a22*b3+a12*b2*a31+b1*a21*a32
-b1*a22*a31-a11*b2*a32-a12*a21*b3
```

```
print(cc1/cc0,cc2/cc0,cc3/cc0)
```

程序名称:D01\_04.lua

```
a=3; b=4; c=6;
p=(a+b+c)/2
s=math.sqrt(p*(p-a)*(p-b)*(p-c))
print(s)
```

程序名称:D01\_05.lua

```
a=1234
b=(a%11)*(a%17)*(a%19)
print(b)
```

### 第 2 章答案

程序名称:D02\_01.lua

```
print("请输入四个数a,b,c,d:")
a,b=io.read("*number","*number")
c,d=io.read("*number","*number")
if a<b then a=b end
if a<c then a=c end
if a<d then a=d end
print(a)
```

程序名称:D02\_02.lua

```
print("请输入四个数a,b,c,d:")
a,b=io.read("*number","*number")
c,d=io.read("*number","*number")
if a<b then a,b=b,a end
```

```
if b<c then b,c=c,b end
if c<d then c,d=d,c end
if a<b then a,b=b,a end
if b<c then b,c=c,b end
if a<b then a,b=b,a end
print(a,b,c,d)
```

程序名称:D02\_03.lua

```
x=1.2345
if math.tan(x)>x-1/3*x^3 then
    print("tan(x)大")
else
    print("tan(x)小")
end
```

程序名称:D02\_04.lua

```
print("请输入x:")
x=io.read("*number")
if x<0 then
    print("-1")
elseif x==0 then
    print("0")
else
    print("1")
end
```

程序名称:D02\_05.lua

```
print("请输入方程的系数a,b,c:")
a,b,c=io.read("*number","*number","*number")
if a==0 then
    if b==0 then
        if c~=0 then
            print("无解!")
        else
            print("全体实数!")
        end
    else
        print("x1=x2="..(-c/b))
    end
else
    dt=b*b-4*a*c
    if dt<0 then
        print("无解!")
    elseif dt==0 then
        print("x1=x2="..(-b/(2.0*a)))
    else
        print("x1="..(-b-math.sqrt(dt))/(2*a))
        print("x2="..(-b+math.sqrt(dt))/(2*a))
    end
end
```

### 第 3 章答案

程序名称:D03\_01.lua

```
print("请输入a,b,c:")
a,b,c=io.read("*number","*number","*number")
while a>0 and b>0 do
```

```

    if a>b then a=a%b else b=b%a end
end
a=a+b; b=c
while a>0 and b>0 do
    if a>b then a=a%b else b=b%a end
end
print(a+b)
程序名称:D03_02.lua
a=0; b=1
while (b-a>0.01) do
    c=(a+b)/2
    d=math.sin(c)+2*c-1
    if d<0 then a=c else b=c end
end
print(a)
程序名称:D03_03.lua
s=0
k=0
for i=1,100 do
    t=1/i
    if k<=1 then s=s+t else s=s-t end
    k=(k+1)%4
end
print(s)
程序名称:D03_04.lua
for i=1,math.huge do
    if i%3==2 and i%5==3 and i%7==2 then
        found=i; break
    end
end
print(found)
程序名称:D03_05.lua
a={7,4,5,6,7,8,9,2,4,5,8,2,5,3,8,4,3,7,6,9}
table.sort(a)
s=0
for i=2,#a-1 do
    s=s+a[i]
end
print(s/(#a-2))
程序名称:D03_06.lua
a={7,4,5,6,7,8,9,2,4,5,8,2,5,3,8,4,3,7,6,9}
table.sort(a)
s=0
for i=4,#a-3 do
    s=s+a[i]
end
print(s/(#a-6))

```

#### 第 4 章答案

```

程序名称:D04_01.lua
for i=1,5 do
    for j=1,6-i do
        io.write("*")
    end
    io.write("\n")
end

```

```

end
    io.write("\n")
end
程序名称:D04_02.lua
for i=1,5 do
    for j=1,i do io.write(" ") end
    for j=1,6-i do io.write("*") end
    io.write("\n")
end
程序名称:D04_03.lua
for i=1,6 do
    for j=1,6-i do io.write(" ") end
    for j=1,2*i-1 do io.write("*") end
    io.write("\n")
end
for i=1,6 do
    for j=1,i do io.write(" ") end
    for j=1,10-2*i+1 do io.write("*") end
    io.write("\n")
end
程序名称:D04_04.lua
for i=1,4 do
    for j=1,5-i do io.write("a") end
    for j=1,i do io.write("*") end
    print()
end
程序名称:D04_05.lua
for i=1,4 do
    for j=1,5-i do io.write("a") end
    for j=1,2*i-1 do io.write("*") end
    for j=1,5-i do io.write("a") end
    print()
end
程序名称:D04_06.lua
for i=1,5 do
    for j=1,5-i do io.write(" ") end
    for j=1,2*i-1 do io.write("*") end
    for j=1,11-2*i do io.write(" ") end
    for j=1,2*i-1 do io.write("*") end
    print()
end
程序名称:D04_07.lua
for i=1,5 do
    for j=1,5-i do io.write(" ") end
    for j=1,2*i-1 do io.write("*") end
    for j=1,11-2*i do io.write(" ") end
    for j=1,2*i-1 do io.write("*") end
    print()
end
for i=1,4 do
    for j=1,i do io.write(" ") end
    for j=1,9-2*i do io.write("*") end
    for j=1,2*i+1 do io.write(" ") end
end

```



```

    for j=1,9-2*i do io.write("*") end
    print()
end
程序名称:D04_08.lua
for i=1,5 do
    for j=1,5-i do io.write(" ") end
    ch=97
    for j=1,2*i-1 do
        io.write(string.char(ch))
        ch=ch+1
    end
    print()
end
for i=1,4 do
    for j=1,i do io.write(" ") end
    ch=97
    for j=1,9-2*i do
        io.write(string.char(ch))
        ch=ch+1
    end
    print()
end
程序名称:D04_09.lua
for i=1,5 do
    for j=1,5-i do io.write(" ") end
    ch=97
    for j=1,2*i-1 do
        io.write(string.char(ch))
        if j<i then ch=ch+1 else ch=ch-1 end
    end
    io.write("\n")
end
for i=1,4 do
    for j=1,i do io.write(" ") end
    ch=97
    for j=1,9-2*i do
        io.write(string.char(ch))
        if j<5-i then ch=ch+1 else ch=ch-1 end
    end
    io.write("\n")
end
程序名称:D04_10.lua
for i=1,9 do
    for j=0,9 do
        for k=0,9 do
            for m=0,9 do
                p=i*1000+j*100+k*10+m
                if p==i^4+j^4+k^4+m^4 then
                    print(i*1000+j*100+k*10+m)
                end
            end
        end
    end
end
end
end

```

```

程序名称:D04_11.lua
for i=2,100 do
    p=0
    for j=2,i-1 do
        if i%j==0 then p=j; break end
    end
    if p==0 then io.write(i,",") end
end
程序名称:D04_12.lua
for x=0,97 do
    for y=0,97 do
        if 3*x+5*y==98 then
            io.write("x=",x,"\t","y=",y,"\n")
        end
    end
end
程序名称:D04_13.lua
s=0
for i=1,7 do
    k=1
    for j=1,i do k=k*j end
    s=s+k
end
print(s)

```

### 第 5 章答案

```

程序名称:D05_01.lua
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30}
k=aa[1]
for i=1,#aa-1 do
    aa[i]=aa[i+1]
end
aa[#aa]=k
for i=1,#aa do
    io.write(aa[i],",")
end
程序名称:D05_02.lua
aa={1,3,4,7,8,10,13,14,16,17,19,21,23,29,30}
for i=1,#aa/2 do
    aa[i],aa[#aa-i+1]=aa[#aa-i+1],aa[i]
end
for i=1,#aa do
    io.write(aa[i],",")
end
程序名称:D05_03.lua
F={1,2,3}
for i=4,12 do
    F[i]=F[i-1]+F[i-2]+F[i-3]
end
for i=1,12 do
    io.write(F[i],",")
end
程序名称:D05_04.lua

```

```

aa={}
for i=2,100 do aa[i]=i end
for i=2,10 do
    if aa[i]~=0 then
        for j=i+1,100,i do aa[j]=0 end
    end
end
for i=2,100 do
    if aa[i]~=0 then io.write(aa[i],",") end
end
程序名称:D05_05.lua
aa={1}
for j=2,11 do aa[j]=0 end
for i=1,10 do
    for j=i,2,-1 do aa[j]=aa[j]+aa[j-1] end
    for j=1,i do
        io.write(string.format("%5d",aa[j]))
    end
    io.write("\n")
end
end

```

### 第 6 章答案

```

程序名称:D06_01.lua
function MinZYS(x)
    if x%2==0 then return 2 end
    for i=3,math.sqrt(x),2 do
        if x%i==0 then return i end
    end
    return x
end
io.write("1=1\n")
for m=2,100 do
    io.write(m,"=")
    n=m
    while n>1 do
        k=MinZYS(n)
        if (m>n) then io.write("*") end
        io.write(k)
        n=(n-n%k)/k
    end
    io.write("\n")
end
程序名称:D06_02.lua
A={1,3,4,7,8,10,13,14,16}

function Inser(x)
    p=0
    for i=#A,1,-1 do
        if A[i]>x then A[i+1]=A[i]
        else p=i; break
        end
    end
    A[p+1]=x
end

```

```

x=io.read("*number")
Inser(x)
for i=1,#A do io.write(A[i],",") end
程序名称:D06_03.lua
A={1,3,3,3,7,7,13,14,15,17,19,21,23,29,30}

function Fount(x)
    if A[#A]<x then return 0 end
    if A[#A]==x then return 1 end
    a,b=1,#A-1
    while b-a>=2 do
        c=((a+b)-(a+b)%2)/2
        if (A[c]>x) then b=c else a=c end
    end
    if (A[a]==x) then return 1 end
    return 0
end

x=io.read("*number")
if Fount(x)==1 then
    print("在数组中")
else
    print("不在数组中")
end
程序名称:D06_04.lua
aa={1}

function Next(i)
    for j=i,2,-1 do aa[j]=aa[j]+aa[j-1] end
end

for j=2,15 do aa[j]=0 end
for i=1,10 do
    Next(i)
    for j=1,i do
        io.write(string.format("%5d",aa[j]))
    end
    io.write("\n")
end
程序名称:D06_05.lua
count=0
for i=2002,2079 do
    if i%400==0 or (i%100~=0 and i%4==0) then
        count=count+1
    end
end
print(count)

```

### 第 7 章答案

```

程序名称:D07_01.lua
A={{1,5,3},{4,2,1},{3,2,0}}
B={{2,1,0},{2,3,1},{3,1,2}}
C={{0,0,0},{0,0,0},{0,0,0}}

```

```

for i=1,3 do
    for j=1,3 do
        for k=1,3 do
            C[i][j]=C[i][j]+A[i][k]*B[k][j]
        end
    end
end
end
for i=1,3 do
    for j=1,3 do
        io.write(string.format("%3d",C[i][j]))
    end
    io.write("\n")
end
程序名称:D07_02.lua
A={{1,5,3},{4,2,1},{3,2,0}}
i=1; j=3; n=#A
for k=1,n do
    A[i][k],A[j][k]=A[j][k],A[i][k]
end
for i=1,n do
    for j=1,n do
        io.write(string.format("%5d",A[i][j]))
    end
    io.write("\n")
end
程序名称:D07_03.lua
A={{1.1,5,3},{4.2,2,1},{3.3,2,0}}
sum=0
for i=1,#A do
    for j=1,#A[i] do
        sum=sum+A[i][j]
    end
end
print("sum="..sum)
程序名称:D07_04.lua
A={{1}}
io.write(string.format("%5d",A[1][1]))
io.write("\n")
for i=2,10 do
    A[i]={}
    A[i][1]=1
    for j=i,2,-1 do
        if not A[i-1][j] then A[i-1][j]=0 end
        A[i][j]=A[i-1][j]+A[i-1][j-1]
    end
    for j=1,i do
        io.write(string.format("%5d",A[i][j]))
    end
    io.write("\n")
end
程序名称:D07_05.lua
A={}

```

```

n=7
for i=1,n do
    A[i]={}
    for j=1,n do A[i][j]=0 end
end
for i=1,n do A[i][i]=i end
for i=1,n-1 do A[i][i+1]=1 end
for i=2,n do A[i][i-1]=2*i-3 end
for i=1,n do
    for j=1,n do
        io.write(string.format("%5d",A[i][j]))
    end
    io.write("\n")
end
程序名称:D07_06.lua
A={}
N=7
for i=1,N do
    A[i]={}
    for j=1,N do A[i][j]=0 end
end
k=1
for n=1,N do
    i=N-n+1; j=1
    while i<=N do
        A[i][j]=k
        i=i+1; j=j+1; k=k+1
    end
end
for i=1,N do
    for j=1,N do
        io.write(string.format("%5d",A[i][j]))
    end
    io.write("\n")
end

```

## 第 8 章答案

```

程序名称:D08_01.lua
ss={}
k=io.read("*number")
while k>0 do
    ss[#ss+1]=k%16
    k=(k-k%16)/16
end
for i=#ss,1,-1 do
    if ss[i]<=9 then
        io.write(string.char(48+ss[i]))
    else
        io.write(string.char(55+ss[i]))
    end
end

```

```

程序名称:D08_02.lua
s1,s2={},{}
k=io.read("*number")
flag=true
while k>0 do
    s2[#s2+1]=k%10
    k=(k-k%10)/10
    if k==0 then flag=false; break end
    s1[#s1+1]=k%10
    k=(k-k%10)/10
end

```

```

if flag then
    while #s1>0 do
        io.write(s1[#s1])
        s1[#s1]=nil
    end
    while #s2>0 do
        io.write(s2[#s2])
        s2[#s2]=nil
    end
    io.write("\n")
else
    while #s2>0 do
        io.write(s2[#s2])
        s2[#s2]=nil
    end
    while #s1>0 do
        io.write(s1[#s1])
        s1[#s1]=nil
    end
    io.write("\n")
end

```

```

程序名称:D08_03.lua
xx={}
print("请输入10-50之间的数x:")
x=io.read("*number")
io.write(x,"!=")
for i=2,x do
    --判断 i 是否为质数,是则压栈
    p=0
    for j=2,i-1 do
        if i%j==0 then p=j; break end
    end
    if p==0 then xx[#xx+1]=i end
end

for i=1,#xx do
    s=0; k=x
    while k>0 do
        k=(k-k%xx[i])/xx[i]; s=s+k
    end
    if (i>1) then io.write("*") end
    io.write(xx[i],"^",s)

```

```
end
```

### 第9章答案

```

程序名称:D09_01.lua
function F(n)
    if n<16 then
        if n<=9 then
            io.write(string.char(48+n))
        else
            io.write(string.char(55+n))
        end
    else
        F((n-n%16)/16)
        if n%16<=9 then
            io.write(string.char(48+n%16))
        else
            io.write(string.char(55+n%16))
        end
    end
end

```

```

k=io.read("*number")
F(k)

```

```

程序名称:D09_02.lua
function F(n)
    if n<=3 then
        return n
    end
    return F(n-1)+F(n-2)+F(n-3)
end

```

```
print(F(12))
```

```

程序名称:D09_03.lua
function F (n)
    if (n==1) then return 1 end
    return n*F(n-1)
end
print(F(5))

```

```

程序名称:D09_04.lua
function F (n)
    if (n==1) then return 10 end
    return F(n-1)+2
end
print(F(5))

```

```

程序名称:D09_05.lua
function F(n,m)
    if n<10 then return m end
    return (F((n-n%10)/10,m+1))
end

```

```

function G(n)
    if n<10 then io.write(n,","); return end
    G((n-n%10)/10)

```

```

    io.write(n%10,",")
end

k=23451;
print(F(k,1))
G(k)
程序名称:D09_06.lua
function JS(n,m)
    if n==0 then
        if m>=1 and m<=3 then return 1 end --赢
    end
    if m==0 then
        if n>=1 and n<=3 then return 1 end --赢
    end
    for i=1,3 do
        if n-i>=0 then
            if JS(n-i,m)==0 then return 1 end
        end
    end
    for i=1,3 do
        if m-i>=0 then
            if JS(n,m-i)==0 then return 1 end
        end
    end
    return 0
end

print(JS(7,10))
程序名称:D09_07.lua
function JS(n,m)
    if n==0 then
        if m>=1 and m<=3 then
            return 1
        end
    end --赢
    if m==0 then
        if n>=1 and n<=3 then
            return 1
        end
    end --赢
    for i=1,3 do
        if n-i>=0 then
            if JS(n-i,m)==0 then
                return 1
            end
        end
    end
    for i=1,3 do
        if m-i>=0 then
            if JS(n,m-i)==0 then
                return 1
            end
        end
    end
end

```

```

    for i=1,3 do
        if m-i>=0 and n-i>=0 then
            if JS(n-i,m-i)==0 then
                return 1
            end
        end
    end
    return 0
end

print(JS(7,10))
程序名称:D09_08.lua
aa={}

function JS(n,m,k)
    if aa[n][m][k]~=0 then
        return aa[n][m][k]
    end
    if n==0 and m==0 then --赢,当两堆为空时
        aa[n][m][k]=1; return 1
    end
    if n==0 and k==0 then --赢,当两堆为空时
        aa[n][m][k]=1; return 1
    end
    if m==0 and k==0 then --赢,当两堆为空时
        aa[n][m][k]=1; return 1
    end
    for i=0,n-1 do
        if JS(i,m,k)==-1 then
            aa[n][m][k]=1; return 1
        end
    end
    for i=0,m-1 do
        if JS(n,i,k)==-1 then
            aa[n][m][k]=1; return 1
        end
    end
    for i=0,k-1 do
        if JS(n,m,i)==-1 then
            aa[n][m][k]=1; return 1
        end
    end
    return -1
end

for i=0,50 do
    aa[i]={}
    for j=0,50 do
        aa[i][j]={}
        for k=0,50 do
            aa[i][j][k]=0
        end
    end
end
end

```

```
print(JS(8,7,9))
```

程序名称:D09\_09.lua

```
function JS(n,m,k)
    if n==0 and m==0 then return 1 end
    if n==0 and k==0 then return 1 end
    if m==0 and k==0 then return 1 end
    for i=0,n-1 do
        if JS(i,m,k)==-1 then return 1 end
    end
    for i=0,m-1 do
        if JS(n,i,k)==-1 then return 1 end
    end
    for i=0,k-1 do
        if JS(n,m,i)==-1 then return 1 end
    end
    return -1
end
-- 纯递归,没有加速 约1分钟
print(JS(8,7,9))
```

### 第 10 章 答案

程序名称:D10\_01.lua

```
io.output("星型图形.txt","w")
for i=1,5 do
    for j=1,5-i do io.write(" ") end
    for j=1,2*i-1 do io.write("*") end
    io.write("\n")
end
io.close()
```

程序名称:D10\_02.lua

```
io.output("数列F.txt","w")
function F(n)
    if n<=2 then return 1 end
    return F(n-1)+F(n-2)
end
```

```
for i=1,12 do
    io.write (F(i)..",")
end
io.close()
```

程序名称:D10\_03.lua

```
aa = {}
sum = 0
-- text.txt 含有十个数,中间用空格连接
fp=assert(io.open("数列1.txt","r"))
for i=1,10 do
    aa[i]=fp:read("*number")
    sum=sum+aa[i]
end
fp:close()
print(sum/10)
```

程序名称:D10\_04.lua

```
fp=assert(io.open("数列3.txt","r"))
s=0
for i=1,4 do
    k=fp:read("*number")
    s=s+k
    for j=1,k do
        k=fp:read("*number") --m用于缓存
    end
end
fp:close()
print("s"..s)
```

## 附录 1 库函数列表

<code>math.abs(x)</code>	-- x的绝对值
<code>math.acos(x)</code>	-- 反余弦(弧度)
<code>math.asin(x)</code>	-- 反正弦(弧度)
<code>math.atan(x)</code>	-- 反正切(弧度)
<code>math.atan2(y,x)</code>	-- 返回y/x的反正切(弧度)
<code>math.ceil(x)</code>	-- 返回大于或等于x的最小整数
<code>math.cos(x)</code>	-- 余弦函数
<code>math.cosh(x)</code>	-- 双曲余弦
<code>math.deg(x)</code>	-- 弧度转成度,返回角x(弧度)中给出的度
<code>math.exp(x)</code>	-- 返回 $e^x$
<code>math.floor(x)</code>	-- 返回小于或等于x的最大整数
<code>math.fmod(x,y)</code>	-- x对y取余数
<code>math.frexp(x)</code>	-- 返回m和e $x=m2^e$
<code>math.ldexp(m,e)</code>	-- 返回 $m2^e$ (e应该是一个整数)
<code>math.log(x)</code>	-- 返回x的自然对数
<code>math.log10(x)</code>	-- 返回x的10为底的对数
<code>math.max(x)</code>	-- 多参数,返回最大值
<code>math.min(x)</code>	-- 多参数,返回最小值
<code>math.modf(x)</code>	-- 返回两个数,x的整数部分,x的小数部分
<code>math.pow(x,y)</code>	-- $x^y$
<code>math.rad(x)</code>	-- 返回角x的弧度
<code>math.random(m,n)</code>	-- 随机数发生器
<code>math.randomseed(x)</code>	-- 初始化随机数种子
<code>math.sin(x)</code>	-- 正弦函数
<code>math.sinh(x)</code>	-- 双曲正弦
<code>math.sqrt(x)</code>	-- 平方根
<code>math.tan(x)</code>	-- 正切
<code>math.tanh(x)</code>	-- 双曲正切
<code>string.byte(s,i,j)</code>	-- 字符串转化为字符代码序列
<code>string.char(...)</code>	-- 数字代码转化为字符
<code>string.dump(f)</code>	-- 返回一个字符串,其中包含的二进制表示给定的函数
<code>string.find(s,pattern,init,plain)</code>	-- 查找第一个匹配模式的字符串
<code>string.format(formatstring,...)</code>	-- 数据格式化成规范格式,第一个参数为格式字符串
<code>string.gmatch(s,pattern)</code>	-- 迭代器函数,每次调用,返回下一个捕获从模式字符串
<code>string.gsub(s,pattern,rep1,n)</code>	-- 搜索字符串s按照模式查找
<code>string.len(s)</code>	-- 返回字符串长度
<code>string.lower(s)</code>	-- 格式化为全部小写字母
<code>string.match(s,pattern,init)</code>	-- 查找第一个匹配模式的字符串
<code>string.rep(s,n)</code>	-- 字符串s,重复n遍
<code>string.reverse(s)</code>	-- 字符串s逆转
<code>string.sub(s,i,j)</code>	-- 返回字符串的子串,从i开始,一直持续到j
<code>string.upper(s)</code>	-- 格式化为全部大写字母

```
table.concat(table,sep,i,j)    -- 字符串数组连接函数,sep为分隔符,从i位置到j位置
table.insert(table,pos,value)  -- 将值向表插入元素到pos位置
table.maxn(table)              -- 表的最大索引号
table.remove(table,pos)        -- 删除表的元素位置pos,将其他元素向前移动
table.sort(table,comp)        -- 对表进行排序,comp为两个元素的比较规则

io.close(file)                -- 关闭文件
io.flush()                    -- 文件刷新
io.input(file)                 -- 输入文件重定义
io.lines(a)                    -- 以读模式打开给定的文件名
io.open(filename,mode)        -- 打开文件
io.output(file)                -- 输出文件重定义
io.popen(prog,mode)           -- 打开文件???
io.read(format)                -- 从文件读入数据
io.tmpfile()                   -- 临时文件
io.type(obj)                   -- 检查文件句柄类型
io.write(...)                  -- 输出写

os.clock()                     -- 返回已经流逝的CPU时间
os.date(format,time)           -- 时间格式
os.difftime(t2,t1)             -- 返回从时间t1到t2的秒数
os.execute(command)            -- 执行命令
os.exit(code)                  -- 调用C函数退出
os.getenv(varname)             -- 返回varname过程环境变量的值
os.remove(filename)            -- 删除文件或目录的名字
os.rename(oldname,newname)     -- 重命名文件或目录命名oldname新名称。
os.setlocale(locale,category) -- 设置程序的当前语言环境
os.time(table)                 -- 返回当前时间时,无参数时,表示给定表指定的日期和时间
os.tmpname()                   -- 返回一个字符串和一个文件名,可以用于一个临时文件

debug.debug()                  -- 进入交互模式,运行输入的字符串
debug.getfenv(o)                -- 返回对象的环境
debug.gethook(thread)           -- 返回当前线程钩子设置
debug.getinfo(thread,func,what) -- 返回一个表与信息功能
debug.getlocal(thread,level,local1) -- 返回的名称和值函数的局部变量
debug.getmetatable(object)      -- 返回给定对象的元表或零
debug.getregistry()             -- 返回注册表中
debug.getupvalue(func,up)       -- 返回的名称和值upvalue与指数函数func
debug.setfenv(object,table)     -- 将给定对象的环境设置为给定的表
debug.sethook(thread,hook,mask,count) -- 设置给定的函数作为一个钩子
debug.setlocal(thread,level,local1,value) -- 分配价值价值函数的局部变量
debug.setmetatable(object,table) -- 将给定对象的元表设置为给定的表
debug.setupvalue(func,up,value) -- 分配价值价值upvalue与指数函数func
```



## 参 考 文 献

- [1] 郝立柱. 汉语文本自动分类 - 市长公开电话数据统计分析 (博士学位论文). 长春: 吉林大学, 2008.
- [2] 郝立丽. 汉语文本数据挖掘 - 基于市长公开电话数据库的统计分析 (博士学位论文). 长春: 吉林大学, 2009.
- [3] 卢开澄. 组合数学算法与分析, 清华大学出版社, 1983.