第 4 讲 循环结构程序设计

（第 4 章 MATLAB 程序流程控制）

目的：

1. 掌握利用 switch 语句实现多分支选择结构的方法。
2. 掌握利用 for 语句实现循环结构的方法。
3. 掌握利用 while 语句实现循环结构的方法。
4. 熟悉利用向量运算来代替循环操作的方法。

---

1. 掌握利用 switch 语句实现多分支选择结构的方法。

练习 1：输入一个百分制成绩，要求输出成绩等级 A、B、C、D、E。其中 90~100 分为 A，80~89 分为 B，70~79 分为 C，60~69 分为 D，60 分以下为 E。

要求：

(1) 用 switch 语句实现。分别输入成绩：–7，9，56，85.6，93，100，109。

(2) 要求输入百分制成绩后要判断该成绩的合理性，对不合理的成绩应输出出错信息。

给出编辑器中的程序（用 switch 语句实现）：

```
sc=input('输入成绩：');
switch floor(sc/10)  %由于 round 后运算的结果是浮点型，这里不能使用
                     %round(sc)
    case {9}         %选择判断量用大括号{}括起，如果是单选项，可以不用{}
        disp('A');   %由于使用 floor 函数，这里不能用 case{9,10}
                     %否则检查不到输入类似 107 分这样的输入错误。
    case {8}         %判断量行不需要用分号结尾
        disp('B');   %执行语句行需要用分号结尾
    case {7}
        disp('C');
    case {6}
        disp('D');
    case num2cell(0:5)
    %num2cell 函数将数字型转为元胞型（即用大括号括起来的类型）
    %此处也可以 case {0,1,2,3,4,5}
          disp('E');
    otherwise
      if sc==100     %用于处理输入 100 分和一百零几分这样的问题
          disp('A');
```

```
    else
          disp('输入成绩出错！');
     end
  end
```

```
%另一种方法。构造 switch 后的表达式，此程序不用 if 语句
sc=input('输入成绩：');
switch floor(sc/10)*(sc<90)+ceil(sc/10)*(sc>=90)  %ceil 向上取整
%  sc<90 的结果是逻辑值 0 或 1，floor(sc/10)*(sc<90)使输入 90 以下分数时取整数部
% 分，ceil(sc/10)*(sc>=90)使得输入[90,100]的值时结果是 10，输入一百零几时运算
% 结果是 11，属于 otherwise
   case {9,10}
       disp('A');
   case {8}
       disp('B');
   case {7}
       disp('C');
   case {6}
       disp('D');
   case num2cell(0:5)          %或用 case {0,1,2,3,4,5}
       disp('E');
   otherwise
       disp('输入成绩出错！');
 end
```

给出命令行窗口中的运行结果：

```
>> Untitled
输入成绩：-7
输入成绩出错！
>> Untitled
输入成绩：9
E
>> Untitled
输入成绩：56
E
>> Untitled
输入成绩：85.6
B
>> Untitled
```

```
输入成绩：93
A
>> Untitled
输入成绩：100
A
>> Untitled
输入成绩：109
输入成绩出错!
>>
```

2. 掌握利用 for 语句实现循环结构的方法。

练习 2：for 循环语句

已知 $\frac{\ln 2}{2}=\frac{1}{3}+\frac{1}{3}\cdot\frac{1}{3^3}+\frac{1}{5}\cdot\frac{1}{3^5}+\cdots+\frac{1}{2n-1}\cdot\frac{1}{3^{2n-1}}+\cdots$，

求 ln2/2 的近似值，当 n 分别取 10、15、20 时，结果是多少？

给出程序及运行结果：

```
n=input('输入整数 n = ');
y1=0;
for k=1:n
    y1=y1+1/(2*k-1)*1/3^(2*k-1); %这里用 for 实现求值
end
m=1:2:(2*n-1);     %这里用向量运算实现求值
y2=(1./m)* ((1/3).^m)'; % 用前行乘以后列来求和，后面行需要转置为列所以'不能省
format long;%15 位有效数字形式输出以便比较精度
[y1,y2,log(2)/2]   %这里比较三种不同方法的求值结果
format short;% 比较完以后将格式设回默认
```

```
命令行窗口
>> Untitled
输入整数 n = 10
ans =
    0.346573590274906    0.346573590274906    0.346573590279973
>> Untitled
输入整数 n = 15
ans =
    0.346573590279972    0.346573590279972    0.346573590279973
>> Untitled
输入整数 n = 20
ans =
    0.346573590279973    0.346573590279973    0.346573590279973
>>
```

练习 3： 求解数列

已知

$$
\begin{cases}
f_1 = 1 \\
f_2 = 0 \\
f_3 = 1 \\
f_n = f_{n-1} - 2f_{n-2} + f_{n-3} \quad n > 3
\end{cases}
$$

求 $f_1 \sim f_{100}$ 中：

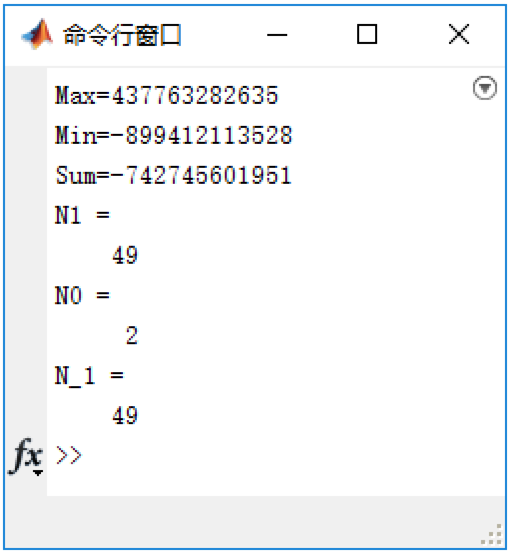
(1) 最大值、最小值、各数之和。

(2) 正数、零、负数的个数。

（注：不能调用 MATLAB 函数 max()、min()、sum()等求各值，要用程序实现。）

给出程序及运行结果：

```
clc;clear ;format short;
f1=1; f2=0; f3=1;
Max=1;Min=0;Sum=2;
N1=2;N0=1;N2=0;
for n=4:100
    f4=f3-2*f2+f1;
    Sum=Sum+f4;%各数之和
    if f4>Max
        Max=f4;%最大值
    elseif f4<Min
        Min=f4;%最小值
    end
    if f4>0
        N1=N1+1;%正数个数
    elseif f4<0
        N2=N2+1;%负数个数
    else
        N0=N0+1;%零个数
    end
    f1=f2;f2=f3;f3=f4;
end
disp(['Max=',num2str(Max)]);
disp(['Min=',num2str(Min)]);
disp(['Sum=',num2str(Sum)]);
N1, N0, N2
```



【提示】

编程方法：

存放数列的变量为 4 个 f1, f2, f3, f4。

初始时，给 f1, f2, f3 赋值，并给各统计变量赋初值。

4:100 循环，按公式计算新值给 f4, 对 f4 进行判断（if 语句），更新各统计变量的值。下次计算前使 f1=f2, f2=f3, f3=f4。

练习 4：亲密数对问题

若两个连续自然数的乘积减 1 是素数，则称这两个连续自然数是亲密数对，该素数是亲密素数。例如，2×3-1=5，由于 5 是素数，所以 2 和 3 是亲密数，5 是亲密素数。求[2,50]区间内：

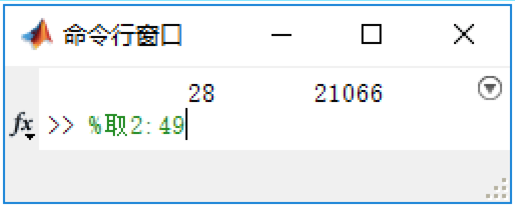
(1)亲密数对的对数。

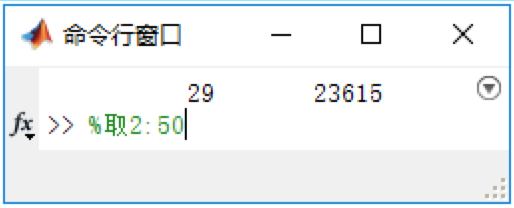
(2) 与上述亲密数对对应的所有亲密素数之和。

给出程序及运行结果：

```
%程序一
clc;clear;format compact;
Num=0; %亲密数对的对数
Sum=0; %亲密素数之和
for k=2:49
    m=k*(k+1)-1;
    if isprime(m)
        Num=Num+1;
        Sum=Sum+m;
    end
end
disp([Num,Sum]);
```

```
%程序二:
clc;clear;format compact;
Num=0; %亲密数对的对数
Sum=0; %亲密素数之和
for k=2:49
    m=k*(k+1)-1;
    for i=2:fix(sqrt(m))+1
        if rem(m,i)==0
            break;
        end
    end
    if i==fix(sqrt(m))+1
        Num=Num+1;
        Sum=Sum+m;
    end
end
disp([Num,Sum]);
%内循环,增加一次循环fix(sqrt(m))+1
%fix(sqrt(m))+1可称为监视哨兵
```





【提示】

（1）如何判断整数 m>1 是素数。若整数 i（$2\le i\le\sqrt{m}$）都不能整除 m，则 m 是素数（m>1 时，$\sqrt{m}<m$）。

（2）可以直接用 `isprime(x)`函数进行判断 x 是否为素数。

3. 掌握利用 while 语句实现循环结构的方法。

练习 5： 用 while 语句求级数部分和

根据 $y=1+\frac{1}{3}+\frac{1}{5}+\cdots+\frac{1}{2n-1}+\cdots$，求：

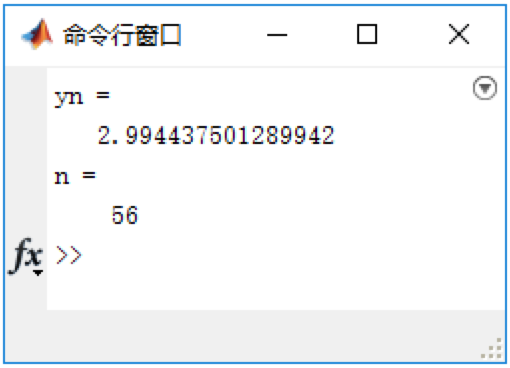
(1) y<3 时的最大 n 值。

(2) 与(1)的 n 值对应的 y 值。

程序及运行结果：

```
%程序一
clc;clear;format long;
n=0; y=0;
while y<3
    n=n+1;
    y=y+1/(2*n-1);
end
yn=y-1/(2*n-1)
n=n-1
format;
```

```
%程序二
clc;clear;format long;
n=0;  y=0;
while y+1/(2*(n+1)-1)<3
    n=n+1;
    y=y+1/(2*n-1);
end
y
n
format short;
```



【提示】

因为 n 值未知，所以用 `while` 来实现较简单。

y 是小于 3 的最大值，但得到的 y 可能是大于 3 的最小值，要对结果的 y，n 进行修正。

用 `format long`，看近似程度。

4. 熟悉利用向量运算来代替循环操作的方法。

练习 6：可以考虑向量运算替代循环语句，

在练习 2 中，可以利用向量运算求级数和：

$$\frac{1}{3}+\frac{1}{3}\cdot\frac{1}{3^3}+\frac{1}{5}\cdot\frac{1}{3^5}+\cdots+\frac{1}{2n-1}\cdot\frac{1}{3^{2n-1}}$$

$$=\left(\frac{1}{1},\ \frac{1}{3},\ \frac{1}{5},\ \cdots,\ \frac{1}{2n-1}\right)*\left(\frac{1}{3^1},\ \frac{1}{3^3},\ \frac{1}{3^5},\ \cdots,\ \frac{1}{3^{2n-1}}\right)'$$

$$=1./(1:2:(2n-1))*\left(\frac{1}{3^1},\ \frac{1}{3^3},\ \frac{1}{3^5},\ \cdots,\ \frac{1}{3^{2n-1}}\right)'$$

练习 2 中的求和正好可以用向量运算来替代。当然不是每个循环都能用向量运算替代。

练习 7：（综合练习）： 迭代过程收敛问题

考虑以下迭代公式：$x_{n+1}=\dfrac{a}{b+x_n}$ 其中 a、b 为正的常数，

迭代的终止条件为迭代次数不超过 500 次或$|x_{n+1}-x_n|\le 10^{-5}$。

背景知识：$x=\dfrac{a}{b+x}\Rightarrow x(b+x)=a\Rightarrow x^2+bx-a=0$，所以迭代过程就是在求此方程的根 $\dfrac{-b\pm\sqrt{b^2+4a}}{2}$ 的近似值。

其迭代求根的原理是：如果 x_0 是方程的准确根，那么 $\dfrac{a}{b+x_0}$ 应等于 x_0，

如果 x_0 是近似根，那么 $\dfrac{a}{b+x_0}$ 应约等于 x_0。

所以当某个值 x_1 能使左右两边近似相等时，该点 x_1 就是近似根。

(1) 编写程序求迭代的结果需要有个迭代终止条件，否则迭代一直进行下去，会陷入死机。此处设置的迭代终止条件为$|x_{n+1}-x_n|\le 10^{-5}$，迭代初值 x_0=1.0，迭代次数不超过 500 次。即当左右两边值差距小于10^{-5}或者迭代次数大于 500 时停止迭代避免死循环。

(2) 如果迭代过程收敛于 r，那么 r 的准确值是 $\dfrac{-b\pm\sqrt{b^2+4a}}{2}$，当($a$,$b$)的值取(1,1)、(8,3)、(10,0.1)时，分别对迭代结果和准确值进行比较。

程序及运行结果：

<table>
<tr>
<td>

```
%程序一 用while实现
format long;
a=input('a = ');  b=input('b = ');
x0=1.0;  x1=a/(b+x0);  n=1;
while n<=500 && abs(x1-x0)>1.0e-5
    x0=x1;  x1=a/(b+x0);  n=n+1;
end
n
x1
```

</td>
<td>

```
a = 1
b = 1
n =
    13
x1 =
   0.618032786885246
r1 =
   0.618033988749895
r2 =
  -1.618033988749895
a = 8
```

</td>
</tr>
</table>

```
r1=(-b+sqrt(b^2+4*a))/2
r2=(-b-sqrt(b^2+4*a))/2
format short;
```

```
%程序二 用for实现
format long;
a=input('a = ');  b=input('b = ');
x0=1.0;
for n=1:500
    x1=a/(b+x0);
    if abs(x1-x0)<1.0e-5
        break;
    end
    x0=x1;
end
n
x1
r1=(-b+sqrt(b^2+4*a))/2
r2=(-b-sqrt(b^2+4*a))/2
format short;
```

```
b = 3
n =
    13
x1 =
   1.701563558633512
r1 =
   1.701562118716424
r2 =
  -4.701562118716424
a = 10
b = 0.1
n =
   424
x1 =
   3.112668149762856
r1 =
   3.112672920173694
r2 =
  -3.212672920173694
>>
```

【提示】

可参考 break、continue——循环的终止和继续。

使用 format long。1.0e-5 表示 10^{-5}，等同 10^(-5)。

注意编程方法，其中用两个变量 x0、x1，x0 存前一个值，计算的新值放入 x1，下次计算前使 x0=x1。不要定义太多无用的变量占用内存空间，这是程序优化原则。

求绝对值的函数为 abs。