《用 TI 图形计算器学编程》—入门篇

要学编程，应该从哪种语言入手？笔者推荐从 BASIC 语言入手，因为它简单易学，在许多国家都作为一种计算机程序的教学语言和入门语言. TI-Nspire™ CX CAS 中文彩屏机，里面的程序语言是 TI-Basic 语言，拥有数量庞大的内置函数，下面我们通过此类 TI 图形计算器，一起走进 BASIC 的学习.

一、BASIC 的故事

BASIC 的诞生，从一个传奇的故事开始.

1964 年，美国达特茅斯大学的 Thomas E. Kurtz(1928－)和 John G. Kemeny（1926.5.31-1992.12.26，原籍匈牙利）在 Fortran II 和 ALGOL 60 的基础上设计了一种新的计算机语言，命名为“Beginner's All-purpose Symbolic Instruction Code”，意为初学者通用符号指令代码，简称 BASIC. 这个简单易学的计算机程序设计语言当时只有 17 条语句，12 个函数和 3 个命令，这就是 BASIC 始祖——Dartmouth BASIC. 第一个 BASIC 程序（实际是一个编译器）于当地时间 1964 年 5 月 1 日凌晨 4 点在一台 GE-265 (IBM 704) 主机中成功运行，操作者是 Mike Busch 和 John McGeachie. 从此 Kurtz 和 Kemeny 作为 BASIC 语言之父被永远载入史册.

BASIC 语言自诞生起就显示出了强大的生命力，各种版本层出不穷. 从 APPLE-Ⅱ机上用的 APPLE SOFT BASIC，到 LASER-310 上用的 MSBASIC，到 IBM-PC 及其兼容机上用的 BASICA 和 GW-BASIC，再到 MS-DOS 上的 MS BASIC、QuickBASIC 和 VisualBasic 1.0， 直至 Windows 下的 Visual Basic 和 Linux 下的 XBasic、YaBASIC 等等，BASIC 无处不在，甚至许多电子游戏机（例如小霸王学习机）和微型电子词典（例如文曲星）中都实现了 BASIC. 从诞生之初，BASIC 语言就以其简单、易学和对硬件要求低等特点受到了广大程序设计人员，特别是电脑初学者的青睐，历经四十载，显示了其顽强的生命力. 如今 BASIC 语言在许多国家都作为一种计算机程序的教学语言和入门语言. 谭浩强教授八十年代所著《BASIC 语言》一书，曾经多次再版，现在的发行量已经超过千万之巨，BASIC 语言在中国有着大量拥趸，许多编程爱好者和专业人士也一直对它情有独钟.

二、TI-BASIC 简介

TI-nspire 系列所使用的 TI-Basic 语言是非常简单的一门语言，同时也是局限性很大的一门语言. 这门语言拥有数量庞大的内置函数，其中相当一部分是数学函数. 在 nspire 上写出的所有程序都需要以函数的形式来运行，这一点会在具体的例子中解释.

创建一个 Nspire 程序，先新建一个计算器后，然后按 menu 9 1 1 即可新建一个程序，如右图.

TI-Nspire 程序分为“程序”和“函数”两类. “程序”能够定义全局变量，可以有任意数量的输出（或者不输出），并且能够调用其它的程序或函数. 而“函数”则只能输出一个结果，不能定义全局变量，也不能调用其它的程序或函数.

运行程序的方法是：在“计算器”页面里输入程序名和左括号，然后输入各项参数，输入右括号后按 Enter 键.

三、程序基础知识

（一）、整体结构与数据类型

Nspire 的程序的整体框架为：

“程序”类	“函数”类
Define [程序名](参数 1，参数 2，…)= Prgm [命令行] EndPrgm	Define [程序名](参数 1，参数 2，…)= Func [命令行] EndFunc

例如 Nspire 上的 Helloworld 程序可以用如下的代码实现：

```
Define hw( )=
Prgm
Disp “Helloworld!”
EndPrgm
```

在 nspire 编程中，数据的类型分为数字、字符串、数组、矩阵、函数和程序. 函数和程序在前文已经介绍过，“数字”指所有的整数和浮点数；“字符串”指在双引号中的数据，如“Helloworld”；“数组”指在花括号中的一组数据，如{1,3,5,7,9}；“矩阵”指以矩阵形式呈现的一组数据. 数据的类型可用 getType()函数判定.

操作提示：在计算器页面，按 menu 9 1 1 新建一个程序，按 menu 2 1 检查语法并保存，按 ctrl tab 在计算窗口与程序窗口跳转. 在计算窗口，按 var 可调用程序（或变量）. 按 ctrl × 可调用字符串符号，按 ctrl) 可调用数组符号“{}”，按 ⬚ 可调用一系列矩阵符号.

（二）、变量的定义，即赋值

Nspire 程序的变量可以通过三种方式定义. 例如把变量 a 定义为 3，有如下三种方式：

方式 1. a:=3　　方式 2. 3→a　　方式 3. Define a=3

除了数字以外，变量也可以用同样的方式定义为其他的数据类型.

此外，与变量操作相关的几个函数也要清楚.

1. Local 函数，其作用是使变量仅在该程序或函数中有效，不作为全局变量.

格式为： Local [变量名 1],[变量名 2],……

2. Delvar 函数，其作用是将变量删除，此操作也同样适用于程序的删除.

格式为：Delvar([变量名])

3. CopyVar 函数，其作用是将变量 1 的内容复制到变量 2，此操作同样适用于程序的复制.

格式为：CopyVar([变量 1]，[变量 2])

（三）、输入输出指令

Nspire 程序有两种输入指令和三种输出指令.

1. Request 输入指令，格式为：Request [“提示”,] 变量名 [,0]

输入指令的第一种为 Request，它要求用户输入的数据为数字，提示部分需要使用字

符串形式. 后面的“0”，表示在输出的结果里不显示这一行的 Request 提示和用户输入值.

例：Request “1+1=?”, ans, 0

它表示若用户输入一个数值，它将被存入变量 ans.

2. RequestStr 输入指令，格式为：RequestStr [“提示”,] 变量名 [,0]

RequestStr 与 Request 的唯一不同在于它对用户输入的值将以字符串形式存入变量.

例：RequestStr “What’s Your Name?”, name, 0

3. Disp 输出指令，格式为：Disp [内容]

Disp 是最基本的输出指令，它在屏幕上显示指定的内容，该内容可以是除程序以外的任意数据类型.

例：Disp “Helloworld”

Disp a+2

4. Tex 输出指令，格式为：Text “[内容]” [,0]

Text 指令会以弹出对话框的形式输出内容，0 的作用与 Request 中的 0 相同. 内容要求为字符串格式.

例：Text “You Win!”, 0

5. Return 输出指令，格式为： Return [内容]

Return 指令为函数专用的输出指令，它使函数返回指定的内容.

例：Return factor(x)

（四）、控制指令

1. If 条件语句

If 句是最常用的条件语句，它有不同的形式.

⑴ If 形式，格式为：If [条件 *a*] [命令 *a*]

最基本的 If 句，它表示：如果条件 *a* 成立则执行命令 *a*. 值得注意的是，命令 *a* 只能有一行.

例：If x>5

x:=x-5

⑵ If-Then-EndIf 形式，格式为：If [条件 *a*] Then [命令 *a*] EndIf

与上一种不同的是，命令 *a* 从 Then 之后延续到 EndIf 之前，没有长度限制.

例：If x>5 Then

x:=x-5

Disp 2x

EndIf

⑶ If-Then-Else-EndIf 形式，格式为：If [条件 *a*] Then [命令 *a*] Else [命令 *b*] EndIf

它表示：如果条件 *a* 成立则执行命令 *a*，否则执行命令 *b*. 在一个 If 句中只能出现一个 Else.

例：If x>5 Then

x:=x-5

Disp 2x

Else

Disp 2x

EndIf

⑷ If-Then-ElseIf-Then-EndIf 形式，格式为：

If [条件 *a*] Then [命令 *a*] ElseIf [条件 *b*] Then [命令 *b*] EndIf

它表示：如果条件 *a* 成立则执行命令 *a*，如果条件 *a* 不成立而条件 *b* 成立则执行命令 *b*. 在一个 If 句中可以出现任意数量的 ElseIf…Then.

例如：If x<3 then
Disp "Too Small"
ElseIf x>3 Then
Disp "Too Big"
ElseIf x=3 Then
Disp "Good!"

2. For 循环语句

For 是 nspire 三种循环句当中的一种.

格式为： For [变量名], [起始值], [结束值], [间隔] [命令行] EndFor

例如：For x,1,5,1
Disp 2x
EndFor

它表示：变量 *x* 的值由 1 增加到 5 且以 1 为间隔递增，*x* 每增加 1 就输出一次 2*x* 的值. 所以这段代码的输出应当为:

2
4
6
8
10

3. While 循环语句

While 也是一种循环句，它的基本格式为： While [条件] [命令行] EndWhile

它表示：如果条件成立则不断重复执行指定的命令，直到条件不成立为止. 所以在 While 句中一定要有一个不断改变的量，否则就会成为一个死循环.

例如：x:=1
While x<6
Disp 2x
x:=x+1
EndWhile

这段语句的功能同上节的 For 语句相同.

4. Loop 循环语句

Loop 是一种死循环句，所以需要与 Exit 指令配合使用.

格式为： Loop [命令行] EndLoop

它的作用就是一直重复执行指定的命令.

例如：x:=1
Loop
Disp 2x

```
    If x≥5
      Exit
    EndLoop
```

这段语句的功能同上两节的语句相同.

5. Try 语句

Try 是非常特殊的一种句法，它的基本格式为：Try [命令 *a*] Else [命令 *b*] EndTry 它表示：如果命令 *a* 能够执行则执行命令 *a*，否则执行命令 *b*.

例如：
```
Request "Enter a number:", num, 0
Try
  If num<0
    Disp –num
Else
  Disp "Invalid Input"
EndTry
```

这段代码要求用户输入一个数字，如果该数小于零，则输出它的相反数. 但是，如果用户输入的不是一个数字而是其它的数据类型或未定义的变量，程序则无法判定其是否小于零，此时便转而执行 Else 后的命令，输出“Invalid Input”.

6. ClrErr 指令

此命令清除当前的错误，并设置了错误的系统变量为零.

7. PassErr 指令

PassErr 将传递“try ... EndTry”的一个错误.

（五）传送指令

1. Lbl 和 Goto

简单来说，Lbl 就是标签，Goto 就是前往标签.

例：
```
Lbl start
x:=x-10
If x>0
Goto start
```

这段代码的功能是将 *x* 值减 10，如果减 10 后的值大于零，则返回上一步再减 10.

2. Exit 和 Stop

Exit 只在循环句中有效，表示跳出该循环. Stop 的作用是直接终止程序（在函数中无效）

例：
```
For i,1,10,1
  x:=10+I
  If int(x/6)=x/6
    Exit
EndFor
```

这段代码表示，在 *i* 从 1 增到 10 的循环中，如果 *i*+10 能被 6 整除则跳出循环.

例：
```
Define example(x)=
Prgm
  If getType(x)≠“NUM” Then
```

```
  Text “Invalid Input”
   Stop
  EndIf
```

这段代码表示，如果用户输入的不是数字，则提示输入无效，直接终止程序的运行.

3. Return

Return 命令将退出一个子程序，并直接返回到父程序的进展.

4. Cycle

Cycle 只在循环句中有效，它表示结束程序循环的某一次周期，遇到时立即开始循环.

例如：

```
Local a
0→a
While a<10
  Disp a
  a+1→a
  If a>5
    Cycle
  a+1→a
EndWhile
```

此代码显示 0，2，4，6，7，8 和 9. 程序运行时，首先在循环体内执行“$a+1\rightarrow a$”两次，从而使“$a+2\rightarrow a$”. 然而，当 a 大于 5 时，第二个“$a+1\rightarrow a$”不会执行，因为周期循环重新启动.

（六）字符串、数组和矩阵相关函数

这一节主要介绍有关字符串、数组和矩阵的主要函数.

1. &

这个字符表示将两个字符串合并. 例：”TI”&”-nspire”，结果为”TI-nspire”.

2. InString()

InString 函数会返回一个（或一段）字符在字符串中出现的位置. 如果该字符多次出现，则返回第一次出现的位置：

例：InString(“cncalc”, “c”)，结果为 1；InString(“TI-nspire”,“ns”)，结果为 4.

3. left()和 right()

如函数名称所表示的，这两个函数分别返回字符串左起或右起的 n 个字符.

例：left(“casio”,3)，结果为”cas”；right(“texas”,3) ，结果为“xas”.

4. dim()

dim()函数返回字符串的长度. 例：dim(“TI Nspire”)，结果为 9.

5. string()

string()函数将一个任意的表达式转换为字符串形式. 例：string(1+2)，结果为“1+2”.

6. expr()

该函数将字符串形式的表达式进行计算，是 string()的逆运算.

例：expr(“1+2”) ，结果为 3.

7. [ ]

方括号的作用是返回数组的第 n 个元素或矩阵的第 i 行 j 列的元素.

例：{1,3,5,7,9}[3]，结果为 5；$\begin{bmatrix}1 & 3\\ 5 & 7\end{bmatrix}$[2　2]，结果为 7.

8. augment()

augment()函数将两个数组进行合并.例：augment({1,3,5},{2,4,6})，结果为{1,3,5,2,4,6}.

9. list▸Mat()

该函数将数组转为每行元素数一定的矩阵. 例：list▸Mat({1,3,5,7},2)，结果为$\begin{bmatrix}1 & 3\\ 5 & 7\end{bmatrix}$

10. Mat▸list()

该函数将矩阵转为数组. 例：Mat▸list($\begin{bmatrix}1 & 3\\ 5 & 7\end{bmatrix}$)，结果为{1,3,5,7}.

四、典型实例

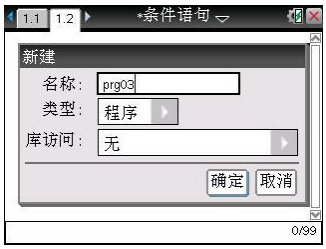
（一）条件语句应用：分段函数计算

例 1 编写一个程序，对于函数 $y=\begin{cases}x & (x<1),\\ 2x-1 & (1\le x<10),\\ 3x-1 & (x\ge 10),\end{cases}$ 输入 x 的值，输出相应函数值.

分析：利用 If-Then-EndIf 语句.

解：第一步 新建一个程序.

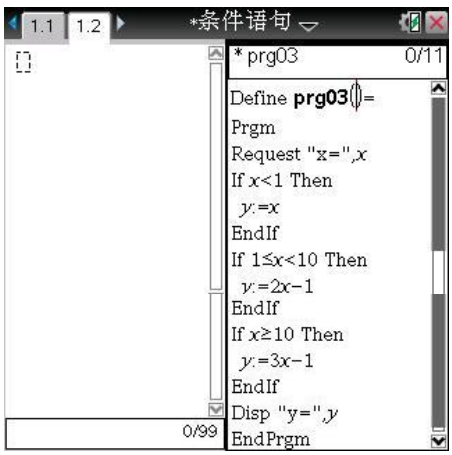
按 [on] [1] [1] 新建一个文档及计算页，按 [menu] [9] [1] [1] 新建一个程序，命名为 prj03.



第二步 输入所编写的程序.

在程序窗口，输入以下程序：

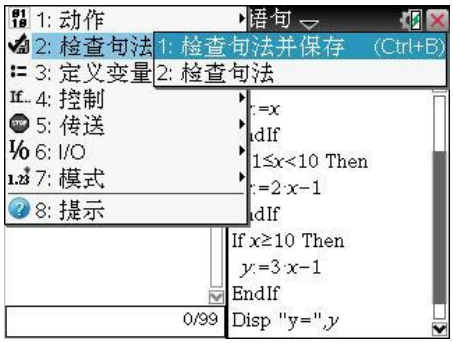
```
Request “x=”,x
If x<1 then
   y:=x
EndIf
If 1 ≤ x<10 then
   y:=2x-1
EndIf
If x ≥ 10 then
   y:=3x-1
EndIf
Disp “y”,y
```



按[ctrl] [x] 可调用字符串符号，按[ctrl] [≠] 可调用定义符 “:=”，按[ctrl] [=]可选择不等号.

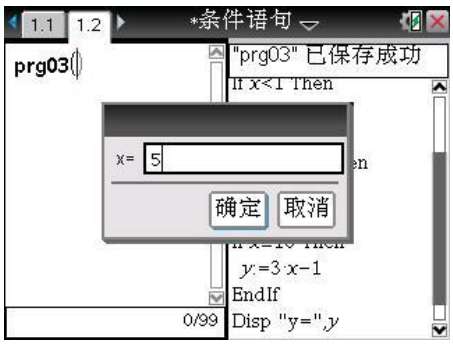
第三步 检查语法，并保存程序.

按 menu 2 1 检查语法并保存，如有错误，则光标停留在错误行.



第四步 运行程序.

按 ctrl tab 跳转到计算窗口，按 var 选择程序（或输入程序名称），按 enter 执行，弹出要求输入变量 x 的窗口，输入一个 x 值，enter.



思考：你能用 ElseIf 语句来编写此程序吗？

（二）循环语句应用；辗转相除法与最大公约数

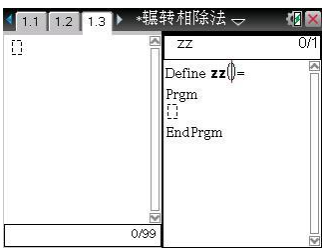
求两个正整数的最大公约数，辗转相除法（出自于活动于公元前 300 年左右的古希腊数学家欧几里得所著的《几何原本》）的算法过程是：用较大的数 m 除以较小的数 n，得到余数 r，即除式 $m = nq + r\ (0 \le r \le n)$，再用除数 n 除以余数 r，得到新的余数. 反复执行这一步，当某步余数为 0 是，该步的除数就是最大公约数.

例 2 用辗转相除法求 8251 与 6105 的最大公约数.

分析：由 8251÷6105=1……2146； 6105÷2146=2……1813；2146÷1813=1……333；1813÷333=5……148； 333÷148=2……37；148÷37=4……0. ∴ 最大公约数为 37.

解：第一步 新建一个程序.

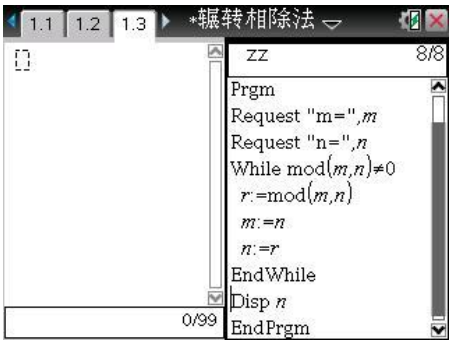
按 on 1 1 新建一个文档及计算页，按 menu 9 1 1 新建一个程序，命名为 zz.



第二步 输入所编写的程序.

在程序窗口，输入以下程序：

```
Request “m=”,m
Request “n=”,n
While mod(m,n) ≠ 0
   r:= mod(m,n)
   m:=n
   n:=r
EndWhile
Disp n
```



按 ctrl × 可调用字符串符号，按 ctrl 可调用定义符“:=”，按 ctrl = 可选择不等号. 其中 mod()为取余函数，格式为：mod(表达式 1,表达式 2).

第三步 检查语法，并保存程序.

按 menu 2 1 检查语法并保存，如有错误，则光标停留在错误行.

第四步 运行程序.

按 ctrl tab 跳转到计算窗口，按 var 选择程序（或输入程序名称），按 enter 执行，弹出要求输入变量 x 的窗口，输入一个 x 值， enter.

思考：还有其它算法求两个正整数的最大公约数吗？

（作者：高建彪　邮箱:dsgjb@163.com，QQ:76456245　2011 年 7 月 12 日完稿于中山市东升高中）

特别说明：本资料的整理完成，感谢两位中学生网友“∫yvre”、“imz”的支持.