

Coding and Calculators

Getting Started WITH THE TI-Nspire

Here, we write a TI-Nspire program to solve a linear equation $ax + b = c$ in three stages: input a , b and c ; solve for x ; and display x .

- On a TI-Nspire, select **New Document** by pressing **1**, by highlighting **New Document** and pressing **enter**, or by positioning the trackpad's mouse on top of **New Document** and pressing the center of the trackpad (see **fig. 1a**). Any of these techniques work in any TI-Nspire menu. Select **Add Calculator** (see **fig. 1b**).
- On the blank **Calculator** screen, use **Menu** → **Functions & Programs** → **Program Editor** → **New**. Then, enter a name for your program (**linear** in this example). These steps lead through **figures 2a, 2b, and 2c** to arrive at **figure 2d**, which has the program editor on the right and a calculator screen on the left.

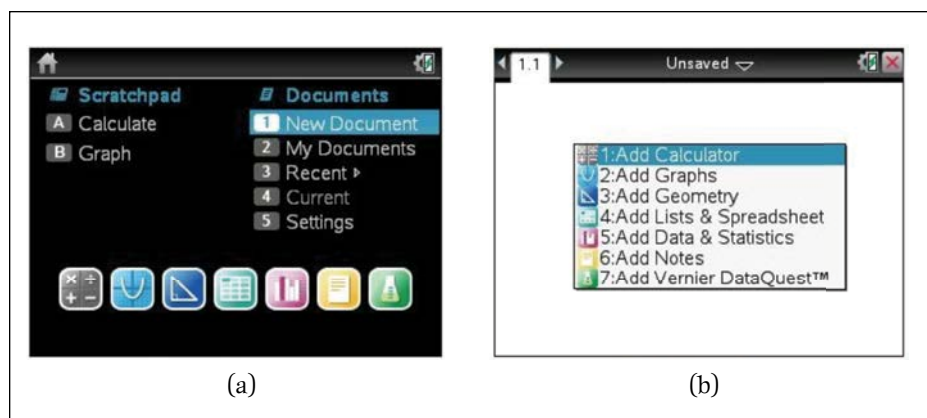


Fig. 1 Creating a program on the TI-Nspire.

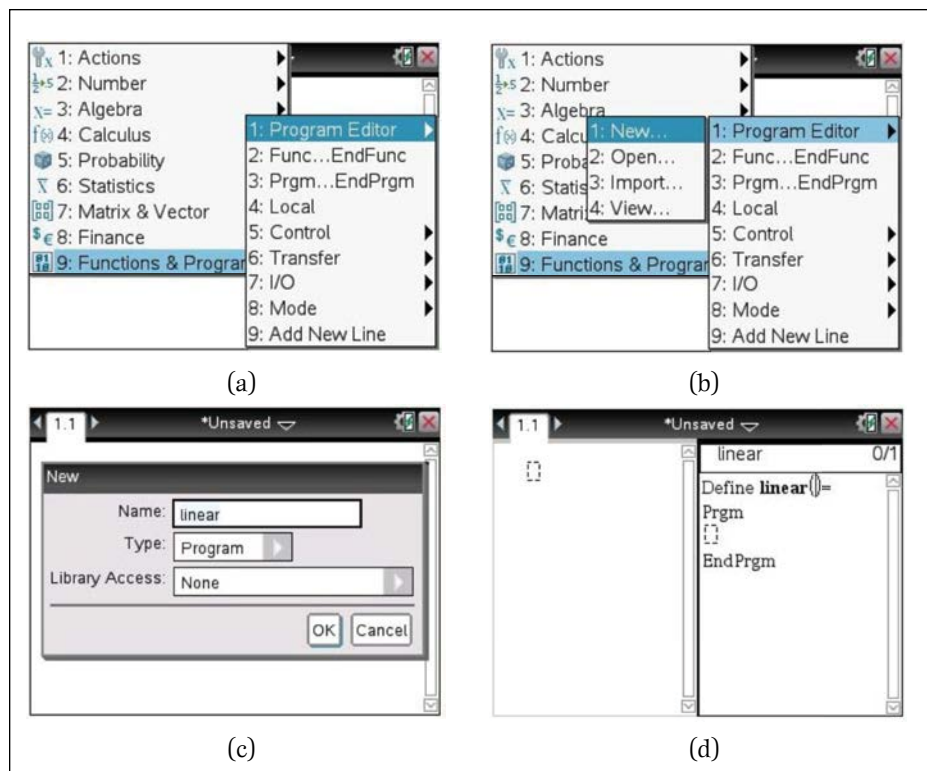


Fig. 2 Invoking the Program Editor.

- Cursor down to the blank line between **Prgm** and **EndPrgm** → **Menu** → **Define Variables** → **Local** (see **fig. 3a**). Then, enter **x**. Use **Local** to declare variables that are not in **Request** or **For** commands (whose variables do not seem to require declaration). Commands can also be found under CATALOG (second icon to the right of the “9,” that looks like an open book).

- Press **enter** for a new line → **menu** → **I/O** → **Request** (see **fig. 3b**) → **ctrl × (“ ”)**. Inside the quotes, enter “a” then press “?” (to right of “G”) and in the popup menu, select “?” (see **fig. 3c**). Cursor to right of the quotes and enter “a” (see **fig. 3d**). When the program is run, it displays a popup box with the first argument as its prompt; the value that has been input is stored in the second argument.

- Enter the next four lines similarly. The symbol to the right of “x” is “:=”. For this symbol press **ctrl :=** (to the right of “9”). **Disp** displays its arguments and is found under **menu** → **I/O** → **Disp** (see **fig. 3b**). Your program should resemble **figure 3d**.

- Before executing any program, the syntax must be checked: **menu** → **Check Syntax & Store** (see **fig. 4a**). Press **ctrl tab** to move from the program editor to the calculator screen. Enter **linear ()** (see **fig. 4b**) to execute the program.

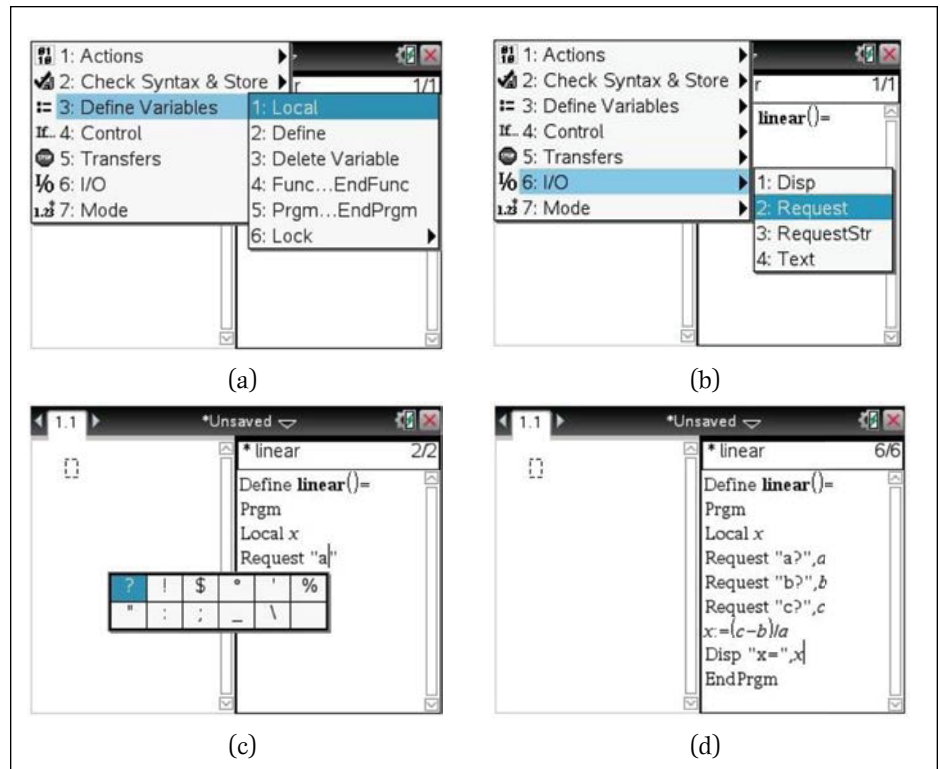


Fig. 3 Entering linear into the Program Editor.

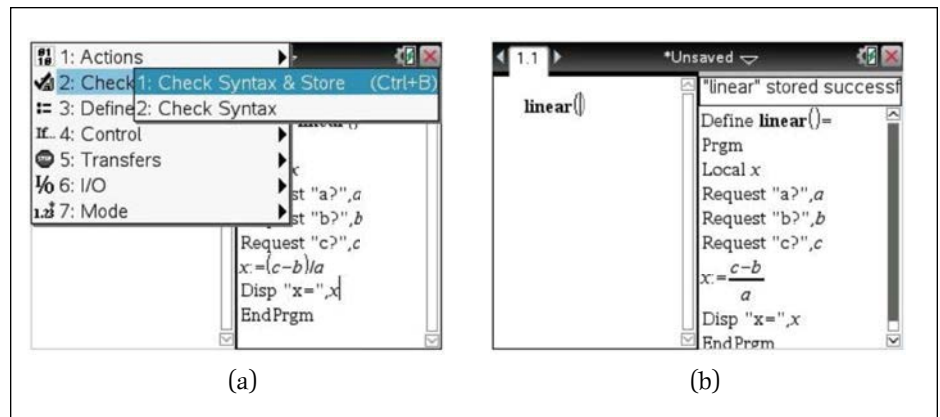


Fig. 4 Checking syntax, executing program linear.

- As the request boxes appear, enter values for a , b , and c (see **fig. 5a**). The input values are echoed on the calculator screen and the value of x is displayed there as well (see **fig. 5b**). Note that x is expressed as a fraction. That is the default mode. If you prefer a decimal, multiply x by 1.0, either in the assignment statement or in the **Disp** statement.
- To save your program: **Doc** → **File** → **Save** or **Save As** (see **fig. 6a**). On the **Save As** screen, the first icon to the right of **Save In** allows you to go up a level in the directory structure; the second allows you to create a new folder (see **fig. 6b**). You can descend a level in the directory structure by highlighting a folder and pressing **enter**.
- To close the Calculator and Program Editor window: **Doc** → **File** → **Close**, or click the Touchpad cursor on the red "X" in the upper right corner.

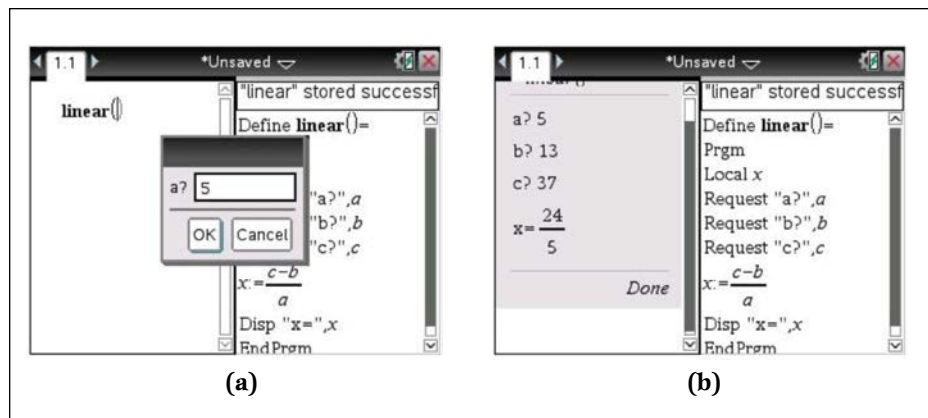


Fig. 5 Input to, and output from, LINEAR.

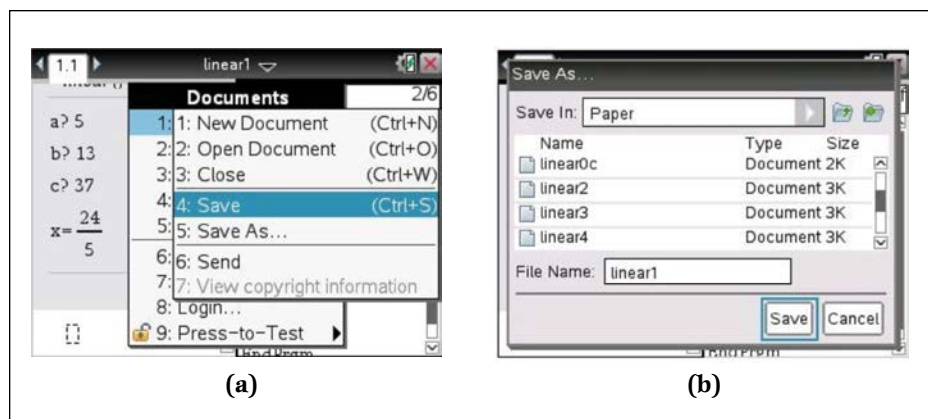


Fig. 6 Saving your program.

Tips for Beginning TI-Nspire Programmers

- To insert a line *before* the line the cursor is on: cursor to the beginning of the line and press **enter**.
- To insert a line *after* the line the cursor is on: cursor to the end of the line and press **enter**.
- The **DEL** key deletes to the left.
- To shift easily from the program editor to the calculator window (and vice-versa): **ctrl tab**.
- The space character is labeled **_** and is found to the right of "Z."
- To exit a document: **Doc** → **File** → **Close**.

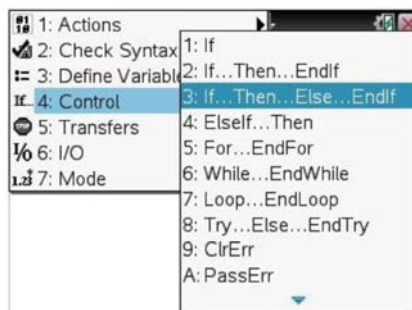
If one enters 0 for a , the program gives x a value of “undef” because it cannot divide by 0. There are several versions of the **If** statement for the TI-Nspire (the TI-Nspire Reference Guide, available from TI’s website, has a listing with all commands and functions and their syntax). **Figure 7a** contains the version of **If** that is appropriate here. Actually, the logic requires an **If** within an **If** (“nested **If**s”) (see **fig. 7b**). To enter the **If** statement: **menu** → **Control** → **If...Then...Else...EndIf** (see **fig. 7c**). The Program Editor enters the framework of the statement (see **fig. 7d**). The user must enter *BooleanExpr*, *Block 1* and *Block 2*.

```
If BooleanExpr Then
  Block 1
Else
  Block 2
Endif
```

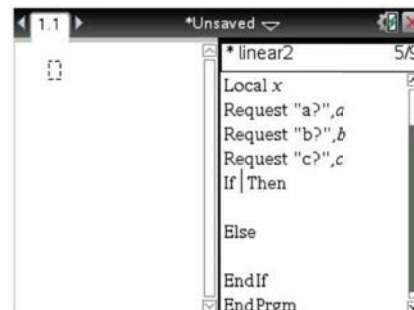
(a)

```
Define linear2()=
Prgm
Local x
Request "a?",a
Request "b?",b
Request "c?",c
If a=0 Then
  If b=c Then
    Disp "All reals"
  Else
    Disp "No sol"
  Endif
Else
  x:=(c-b)/a
  Disp "x=",x
Endif
EndPrgm
```

(b)



(c)



(d)

Fig. 7 If’s syntax / locating If command / entry of If into linear2.

Figure 8 presents a program for the TI-Nspire that is equivalent to the TI-84 version shown in **figure 6** of the main article.

Figure 9 presents a program for the TI-Nspire that is equivalent to the TI-84 version shown in **figure 8** of the main article.

The round function in **Figure 9** rounds its argument to 3 decimal places; without the rounding, the imaginary decimal may run off the right edge of the screen.

```
Define tryhand()=
Prgm
Local a,b,c,d,x
Disp "ax-b=cx-d"
Request "a?",a
Request "b?",b
Request "c?",c
Request "d?",d
If a=c Then
  If b=d Then
    Disp "All reals"
  Else
    Disp "No sols"
  EndIf
Else
  x:=(b-d)/(a-c)
  Disp "x=",x
EndIf
EndPrgm
```

Fig. 8 A TI-Nspire program solves $ax - b = cx - d$.

```
Define quadratic2()=
Prgm
Local a,b,c,d,x,x1,x2
Disp "a*x^2+b*x+c=0"
Request "a?",a
Request "b?",b
Request "c?",c
If a=0 Then
  If b=0 Then
    If c=0 Then
      Disp "All reals"
    Else
      Disp "No sol"
    EndIf
  Else
    x:=((-c)/(b))
    Disp "x=",x
  EndIf
Else
  d:=b*b-4*a*c
  If d=0 Then
    x:=((-b)/(2*a))
    Disp "x=",x
  Else
    If d>0 Then
      x1:=((-b+√(d))/(2*a))
      x2:=((-b-√(d))/(2*a))
      Disp "x1=",x1
      Disp "x2=",x2
    Else
      x1:=round(((-b)/(2*a)),3)+round(((√(-d))/(2*a)),3)*i
      x2:=round(((-b)/(2*a)),3)-round(((√(-d))/(2*a)),3)*i
      Disp "x1=",x1
      Disp "x2=",x2
    EndIf
  EndIf
EndIf
EndPrgm
```

Fig. 9. A TI-Nspire program solves $ax^2 + bx + c = 0$.