

Reference: Useful Menus

While editing a program, navigate to any of these menus to add certain functions or variables to the program.

PRGM	For all of your programming needs.
Catalog (2 nd 0)	The most useful menu! Contains <i>literally everything</i> , in alphabetical order! Type a letter to jump to that section, or press the UP arrow to go straight to punctuation marks.
VARs	Variables! Did you know you can store strings in V_1 or $Str1$?
MODE	Insert commands into your program to change the mode. Need degrees instead of radians? Need Function mode instead of Parametric mode?
LIST (2 nd STAT)	List names, list operations.
DRAW (2 nd PRGM)	Drawing functions, pixel manipulation, picture storage/recall.
MATRIX (2 nd \times^{-1})	Matrix names, matrix math functions.
MEM (2 nd +)	On a TI-84, use this to Archive or UnArchive programs. Note, the TI-83 has two types of memory: RAM and ROM. RAM is fast, but is cleared when the calculator crashes – archive your programs into ROM to save them against crashes! For the TI-83, which only has RAM, just be sure to keep your batteries fresh.
FORMAT (2 nd ZOOM)	Modify various formatting parameters of the graphing window.
ZOOM	Modify the zoom of the graphing window. Note: to set X_{min} and its friends, instead use VARs .
ANGLE (2 nd APPS)	Convert from degrees to radians.
STAT	Statistics tools: regressions and statistical tests.
DISTR (2 nd VARs)	Statistics distributions.
STATPLOT (2 nd $Y=$)	Control statistics plots. For example: $\text{Plot1}(\text{Scatter}, L1, L2, \dots)$

Other Useful Tricks

2nd-ALPHA for alpha-lock: Types in all caps; also, scroll up/down the program much faster.

2nd-DEL for insert mode: Insert characters instead of deleting; this is how you insert newlines.

CLEAR to clear entire lines. DEL to delete single characters.

L to make custom-named lists: e.g. L_{STUFF} , L_X , L_Y — these behave exactly like the standard lists L_1 , L_2 , etc. Find this L in the CATALOG.

Do you have $\text{Ctrl}9\text{Help}$ in your APPS? If so, turn it on! Then, you can press the [+]
button when hovering over any function to read about the arguments that function uses.

Example Programs

PRGMSQUAREIT

```
:Input "X? ", X
:Disp X2
```

PRGMQUADR

```
:Input "A? ", A
:Input "B? ", B
:Input "C? ", C
: $(-B + \sqrt{B^2 - 4AC}) / (2A) \rightarrow X$ 
:Disp X
: $(-B - \sqrt{B^2 - 4AC}) / (2A) \rightarrow X$ 
:Disp X
```

PRGMEVIL

```
:While 1
  :Input "", X
  :Disp X+randInt(-1,1)
:End
```

PRGMDBLTHINK

```
:While 1
  :Input "", X
  :If X=4
  :Then
    :5→X
  :End
  :Disp X
:End
```

PRGMLISTSYAY

```
: {0,0,0,0,0} →L1
: 7→L1(1)
: 3→L1(6)
:Disp L1
:Disp dim(L1)
```

PRGMMYSTERY

```
: {1,1} →L1
: 3→N
:While N≤100
: L1(N-1)+L1(N-2)→L1(N)
: N+1→N
:End
```

A Guessing Game

This program uses recursion – specifically a technique called *binary search* – in order to guess your number as fast as it can.

```
Pr9mGUESSER
:1→L //low bound
:100→H //high bound
:Disp "THINK OF A NUMB-"
:Disp "ER BETWEEN 1 AND
:Disp "100. DON'T TELL!"
:Pause
:Pr9mGUESSFN

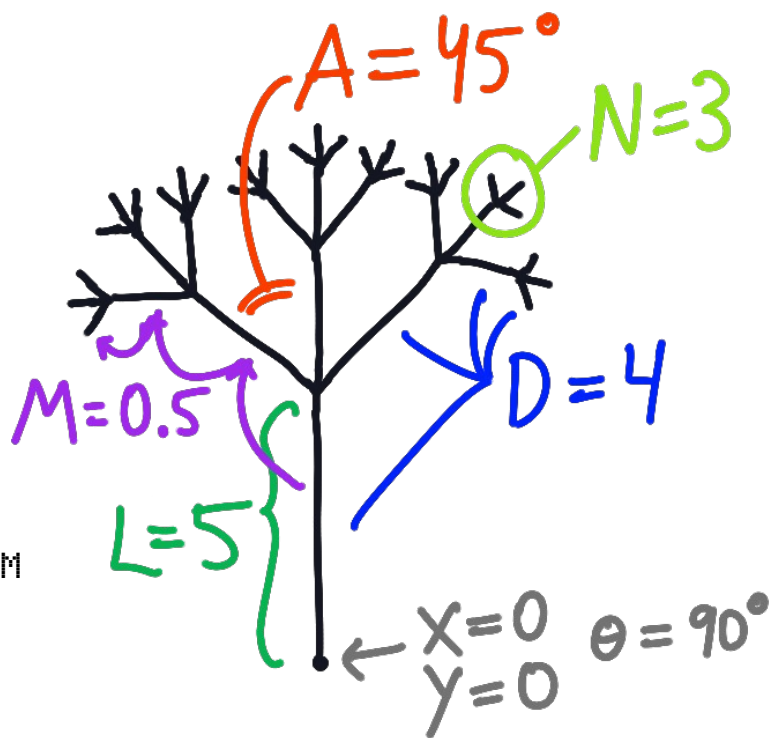
Pr9mGUESSFN
:Disp ""
:If L=H
:Then
:Disp "YOUR NUMBER IS:"
:Disp L
:Else
:int((L+H)/2)→G //make a guess
:Disp "WHAT ABOUT THIS?"
:Disp "MY GUESS IS:"
:Disp G
:Pause
:Menu("YOUR NUMBER IS", "GREATER", A,
"SMALLER", B, "EQUAL", C)
:Lbl A
:G+1→L //check the top half
:Pr9mGUESSFN
:Goto 0
:Lbl B
:G-1→H //check the bottom half
:Pr9mGUESSFN
:Goto 0
:Lbl C
:Disp "YAY! I GUESSED YOUR NUMBER!"
:Goto 0
:Lbl 0
:End
```

Fractal Drawing

```

Pr9mTREEMAIN
: -9.4 → Xmin
: 9.4 → Xmax
: 0 → Ymin
: 12.4 → Ymax
: AxesOff
: ClrDraw
: Degree
: Input "DEPTH? ", D
: Input "LENGTH? ", L
: Input "ANGLE? ", A
: Input "MULTIPLIER? ", M
: Input "BRANCHES? ", N
: 0 → X
: 0 → Y
: 90 → θ
: Pr9mDRAWTREE

```



```

Pr9mDRAWTREE
: X → L1(dim(L1)) // Push all the old variables
: Y → L1(dim(L1))
: θ → L1(dim(L1))
: L → L1(dim(L1))
: D → L1(dim(L1))
: I → L1(dim(L1))
: Line(X, Y, X+Lcos(θ), Y+Lsin(θ))
: If D > 0 // This is very important!
: Then // Otherwise we loop forever!
: D-1 → D
: X+Lcos(θ) → X // New D, X, Y, L, and θ
: Y+Lsin(θ) → Y
: ML → L
: θ-NA/2 → θ
: For(I, 1, N)
: Pr9mDRAWTREE // RECURSION!
: θ+A → θ
: End
: End
: L1(dim(L1)) → I // Pop all the old variables...
: L1(dim(L1)-1) → D
: L1(dim(L1)-2) → L
: L1(dim(L1)-3) → θ
: L1(dim(L1)-4) → Y
: L1(dim(L1)-5) → X // ...and delete the end of L1
: dim(L1)-6 → dim(L1)

```

In-Class Exercises

1. Write a program that asks the user for the two side lengths of a right triangle – named A and B – then computes the length of the hypotenuse, $\sqrt{A^2+B^2}$. Save it to the variable C, and also Disp it.
2. Write a program that calculates the sum $1 + 2 + \dots + 100$. Save it to S, and also Disp it.
3. Write a program that puts the first 10 odd numbers (1, 3, ...) into L_1 and the first 10 even numbers (2, 4, ...) into L_2 .

Challenges to Try

General Programming Practice

- Make yourself more math tools, like the `QUADR` program we wrote! What's a problem that you need to quickly solve a lot in your classes?
- For example: Make a program that takes in certain facts about a triangle – its side lengths, for example – and computes the angles, the area, and other facts. You'll want to use trigonometry, the law of sines, the law of cosines, Heron's formula, etc.
- Another example: Make a program that converts any base-10 number into a base-2 number.
- Make a math quiz game! Quiz yourself on multiplying numbers, squaring numbers, finding sines and cosines of simple angles, and more.

Guessing Game Program

- Modify our guessing game program so that it counts the number of guesses it takes.
- Modify our guessing game program so that it lets you specify the values of `L` and `H`.
- Math problem for you: If there are N possible numbers that you can choose (in this case, $N=100$), how many guesses does the program need to find your number?

Fractal Program

- Modify the fractal-drawing program so that draws dot-marks at the very tips of each branch.
- Can you draw any other fractals using recursion? Sierpinski triangle? Koch curve? Dragon curve?
- Currently it uses *depth-first search*: when we draw out the branches, it goes first to the deepest twigs, before coming back to fill in the next of the larger branches. If the depth is large, it takes a while before we can see any fractal at all! Try modifying it so that it instead uses *breadth-first search*: for example, if $N=3$, first it does the largest 3 branches, then the next largest 9, then the next largest 27, and so on.

Cause-the-Errors Challenges

These are some of the errors you might encounter if you make a mistake while writing a program. Here's a fun reversal on the usual ways of the world: try making programs to *cause* each of these errors! How many can you make?

ERR: ARCHIVED
ERR: ARGUMENT
ERR: BAD GUESS
ERR: BOUND
ERR: BREAK
ERR: DATA TYPE
ERR: DIM MISMATCH
ERR: DIVIDE BY 0
ERR: DOMAIN
ERR: ILLEGAL NEST
ERR: INCREMENT
ERR: INVALID
ERR: INVALID DIM
ERR: ITERATIONS
ERR: LABEL
ERR: MODE
ERR: NO SIGN CHNG
ERR: NONREAL ANS
ERR: OVERFLOW
ERR: SINGULAR MAT
ERR: SINGULARTY
ERR: STAT
ERR: STAT PLOT
ERR: SYNTAX
ERR: TOL NOT MET
ERR: UNDEFINED
ERR: WINDOW RANGE
ERR: ZOOM

Consult <http://tibasicdev.wikidot.com/errors> for hints.

Internet Resources

Turing Machines

Lego Turing machine: <http://vimeo.com/44202270>

Minecraft Turing machine: <https://www.youtube.com/watch?v=1X21HQphy6I>

Conway's Game of Life Turing machine: <https://www.youtube.com/watch?v=My8AsV7bA94>

TI-84 Programming

On making a simple program: http://www.manning.com/mitchell/PTIsample_ch01.pdf

TI-84 features: http://education.ti.com/sites/US/downloads/pdf/TI83-84_reference_guide.pdf

Wikibooks tutorial: http://en.wikibooks.org/wiki/TI-Basic_84_Programming

ticalc.org tutorial: <http://www.ticalc.org/programming/columns/83plus-bas/cherny/>

Fractals and Recursion

Draw fractals: <http://recursivedrawing.com/>

Draw fractals: <http://sciencevmagic.net/fractal>

“Stack Hacking: Programming Recursion in your TI-84.”

Chelsea Voss, Splash 2014

csvoss@mit.edu