A Fast Logarithm Algorithm

While researching using the Internet on the different algorithms used for calculating logarithms, I found an article in PDF form that gave an extremely fast binary (number base 2) algorithm. It was intriguing because it only used the four basic functions of math: addition, subtraction, multiplication and division. I experimented and found out that with minimal modifications the algorithm could be modified to calculate logarithms in ***any*** number base. The results appear to be quite accurate to many decimal places. Below is the link to the online PDF, the original algorithm, and my modifications (in pseudo code) which allow for easy transcription into any computer coding language (I coded my version as a standalone Windows executable using QB64).

****

A Fast Binary Logarithm Algorithm

IEEE SIGNAL PROCESSING MAGAZINE [140] SEPTEMBER 2010

http://www.claysturner.com/dsp/BinaryLogarithm.pdf

We list the binary (Base 2) logarithm algorithm's steps as follows:

1) Initialize result to 0:
y = 0

2) Initialize mantissa-bit decimal value to 0.5:
b = 1 / 2

3) While x < 1, x = 2 * x, y = y - 1

4) While x ≥ 2, x = x / 2 , y = y + 1

5) Go to Step 3 and repeat until:
1 ≤ x < 2

6) Square:
x = x * x

7) If x ≥ 2, x = x / 2, y = y + b

8) Scale for next bit:
b = b / 2

9) Go to Step 6 and repeat until desired number of mantissa bits are found

10) Results:
log(x) = y

REFERENCE
[1] D. E. Knuth, The Art of Computer Programming:
Seminumerical Algorithms, 2nd ed. Reading, MA:
Addison-Wesley, vol. 2, 1981, pp. 441–466

My Notes & Modifications for ANY number base

Note: The four variables below are all double precision

Change x to the number chosen for User Input: **WhatNumber** (Numbers > 0, decimals allowed)
Change 2 to the base chosen for User Input: **WhatBase** (Bases > 1, decimals allowed)
Change b to the mantissa of the log: **Mantissa**
Change y to the log results: **TheLog**

Note: *I inserted the value of these two Number Bases in my source code:*
(Base e) **WhatBase** = 2.71828182845904#
(Base pi) **WhatBase** = 3.14159265358979#

*** Start of Pseudo Code ***

1) Initialize result and integer counter to 0:
TheLog = 0
Count% = 0

2) Initialize mantissa-bit decimal value to 0.5:
Mantissa = 1 / 2

3) While WhatNumber $<$ 1, WhatNumber = WhatBase * WhatNumber, TheLog = TheLog - 1

4) While WhatNumber $\geq$ WhatBase, WhatNumber = WhatNumber / WhatBase , TheLog = TheLog + 1

5) Go to Step 3 and repeat until:
1 $\leq$ WhatNumber $<$ WhatBase

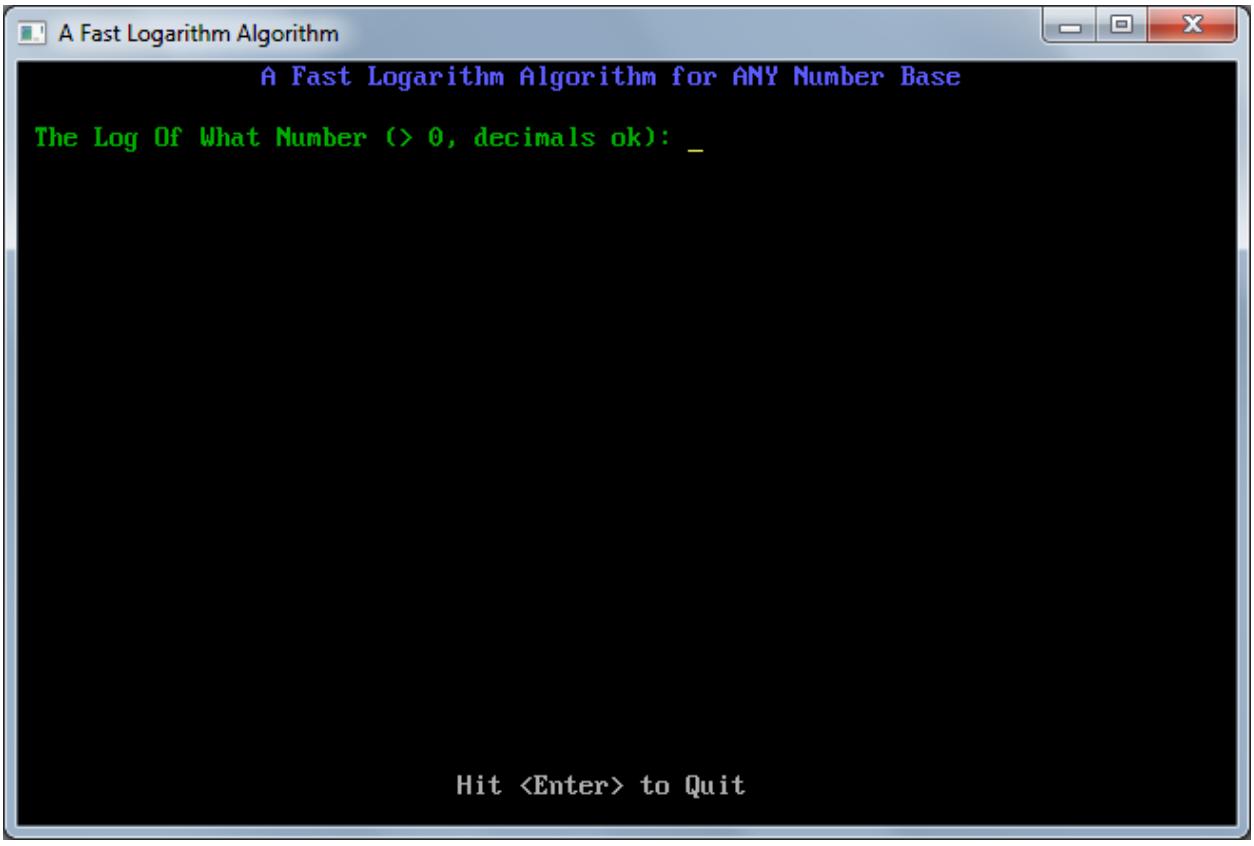
6) Square:
WhatNumber = WhatNumber * WhatNumber

7) If WhatNumber $\geq$ WhatBase, WhatNumber = WhatNumber / WhatBase, TheLog = TheLog + Mantissa

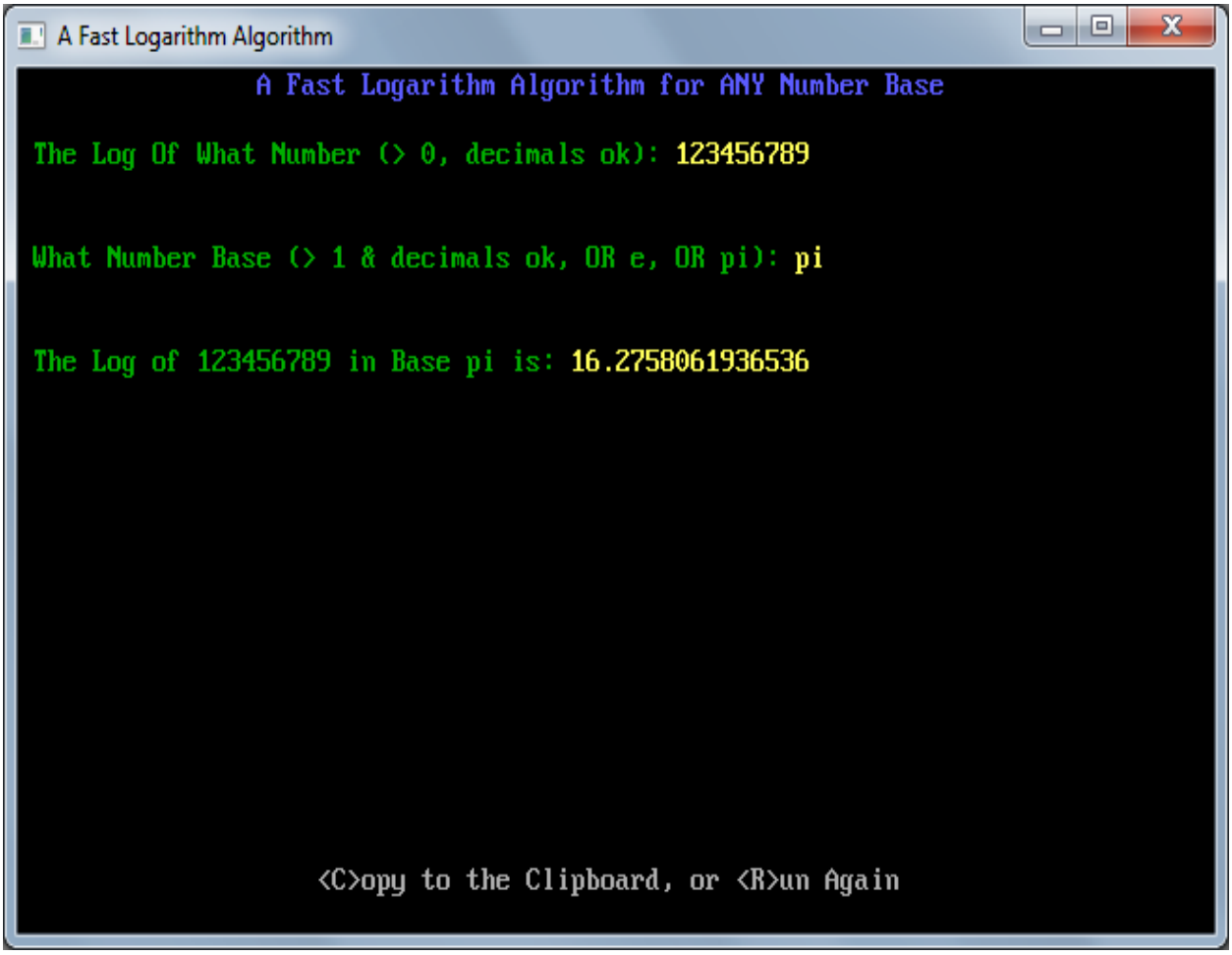
8) Scale for next bit:
Mantissa = Mantissa / 2

9) Go to Step 6 and repeat until desired number of mantissa bits are found using a counter (**Count%**)

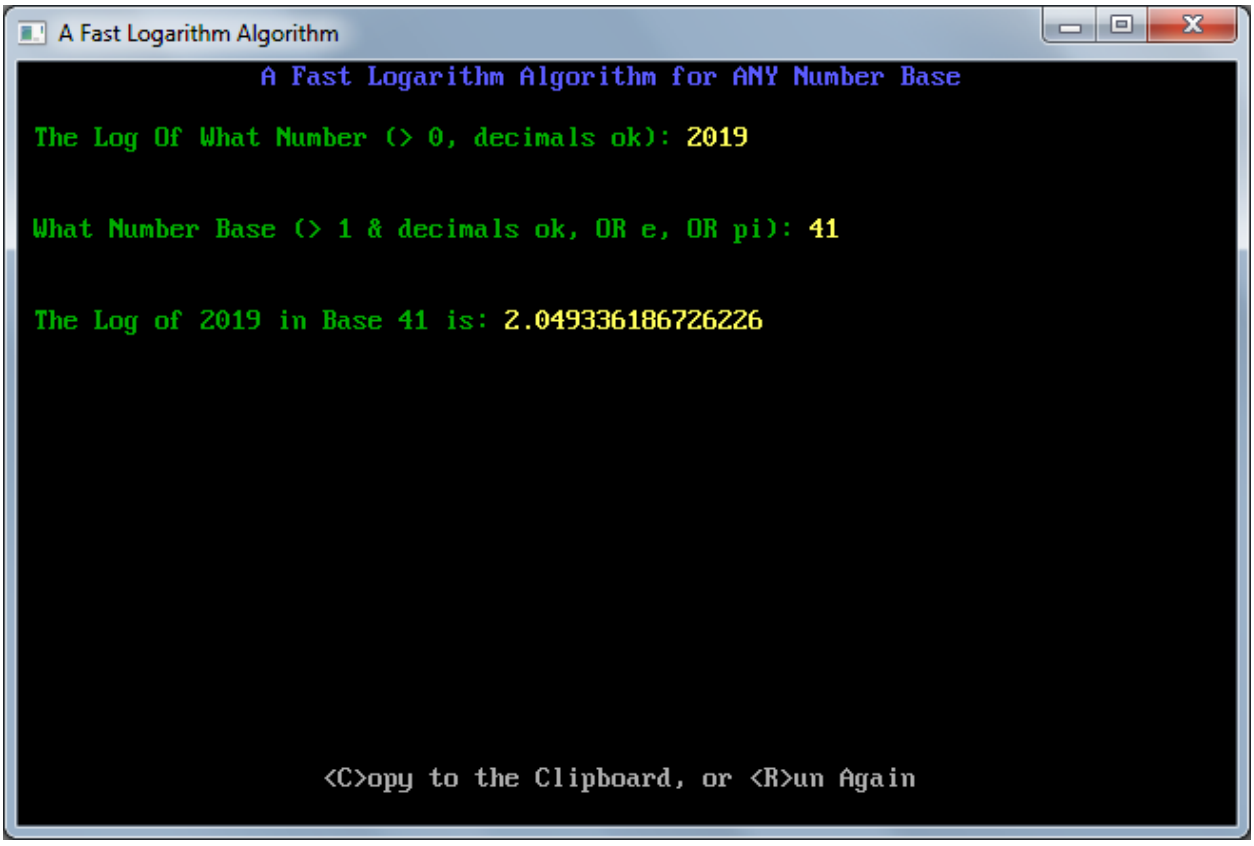
10) Results:
log(WhatNumber) = TheLog



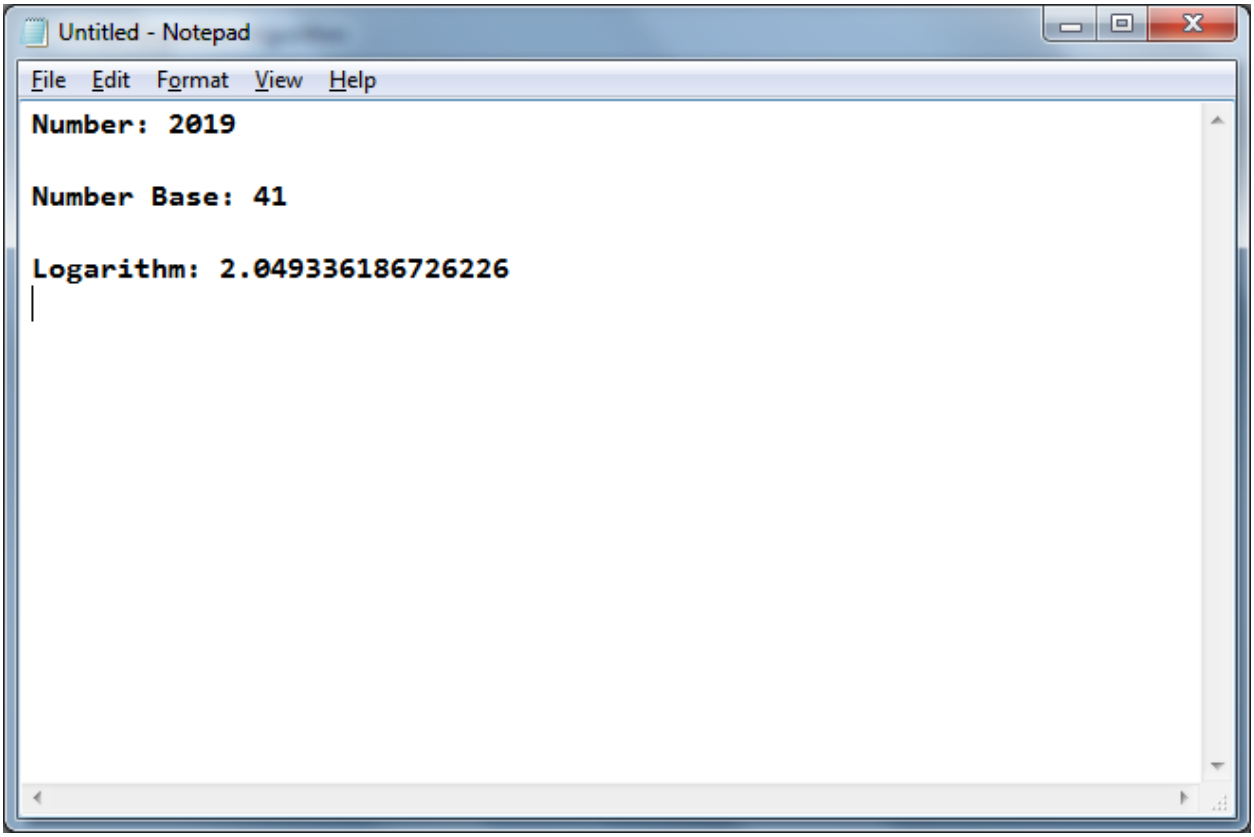
The opening screen (***all*** screens have flashing prompts at the bottom)



After two entries, the results



An arbitrary number base



Results of the last screen copied to the Windows Clipboard and pasted into Notepad