How to program in colorForth cf2019

Contents

Introduction

This is a tutorial on how to program in colorForth under Windows, using the cf2019 system that I distribute and support.

colorForth is a dialect of the Forth programming language, both of which languages were invented by Charles H. "Chuck" Moore ; - Forth around 1968, and colorForth in the late 1990's.

The file cf2019_colorForth.pdf gives a more detailed explanation.

colorForth is not like anything else.

When you press a key in colorForth something happens. Exactly what happens depends on the current state of the system, and this system state is displayed on the screen. In the abstract, this is true of every computer program – the difference with colorForth is that it is complete – there is nothing else going on.

cf2019 is and always will be work in progress. I hope to add version control, strings etc. etc. but without losing colorForth's amazing simplicity.

My intention in supporting colorForth is to make it accessible to as many people as possible.

Feedback welcome howerd@inventio.co.uk www.inventio.co.uk.

Have fun!

Howerd Oakford

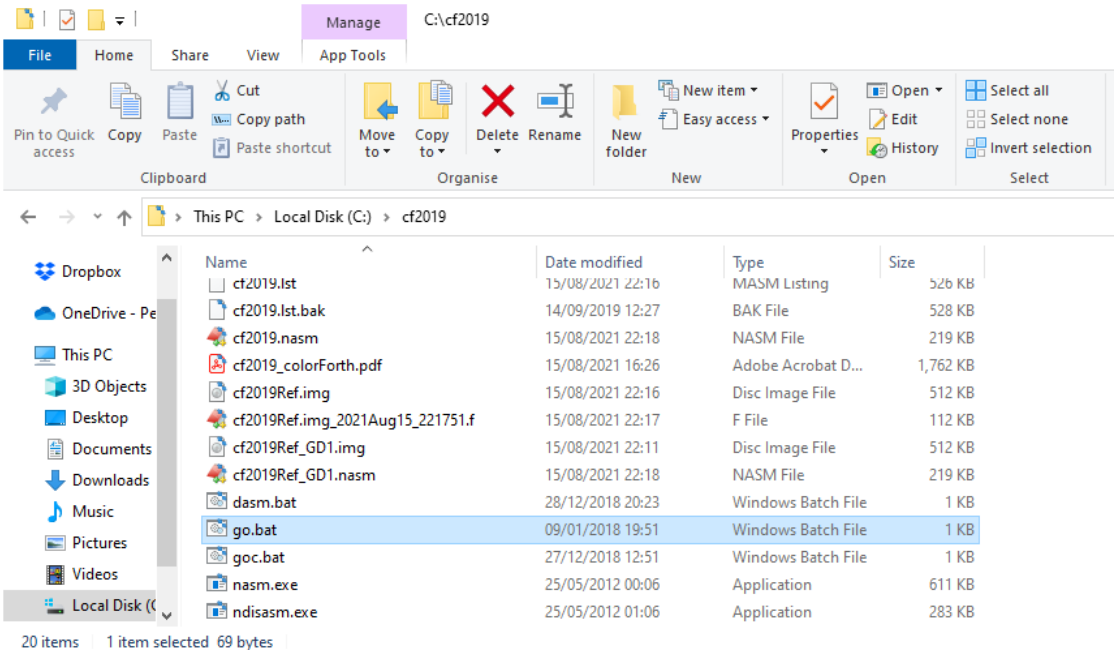
Step 1 Download and run colorForth using the file “cf2019 Issue 2021Aug15.7z” from :

http://www.inventio.co.uk/colorforth/ or by clicking here :

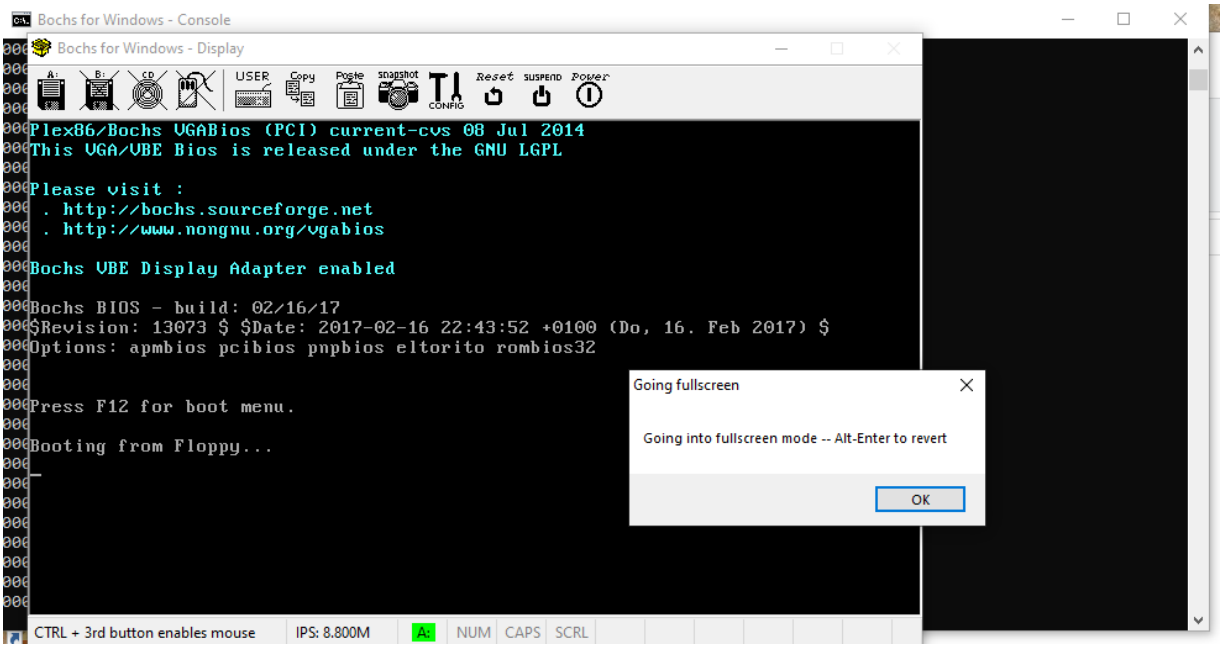
https://www.inventio.co.uk/colorforth/cf2019%20Issue%202021Aug15.7z

Unzip the file on a computer running Windows, for example into the folder cf2019.

Double click on **go.bat** .

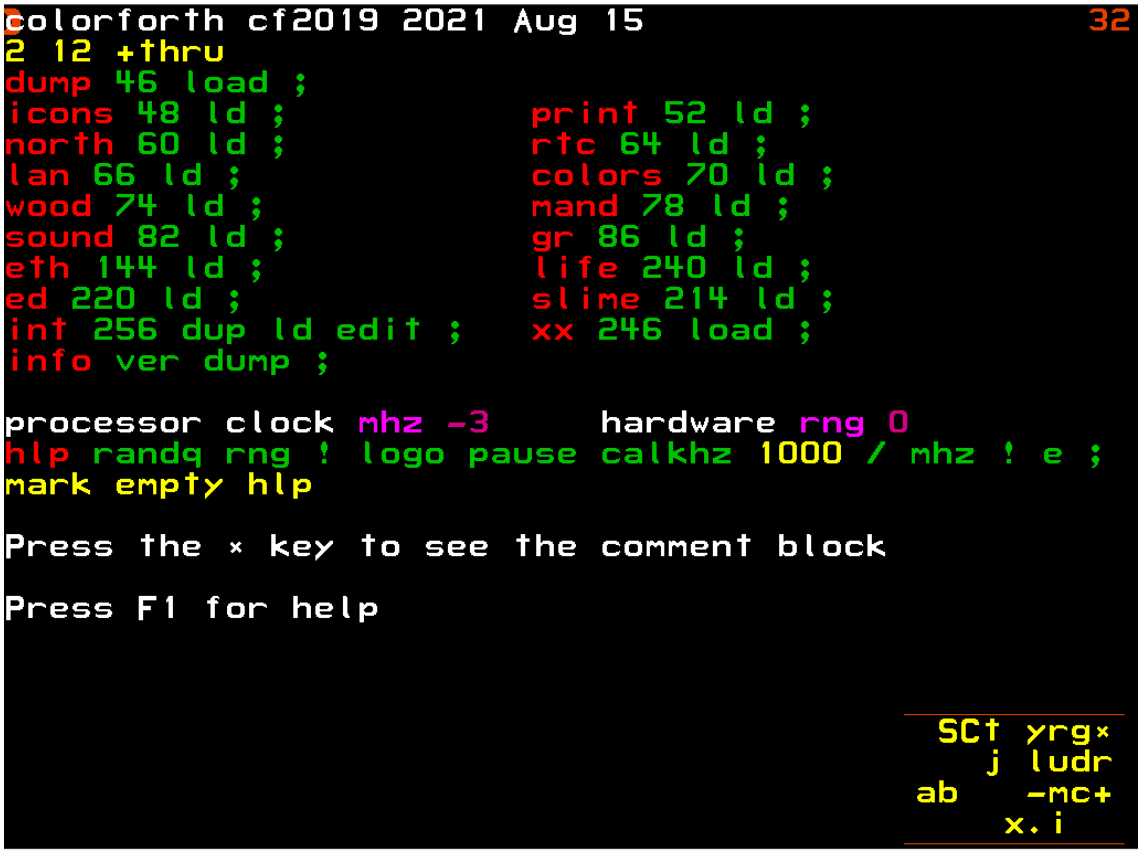


Press the Enter key on the keyboard or click on OK to allow Bochs to run in full screen mode :

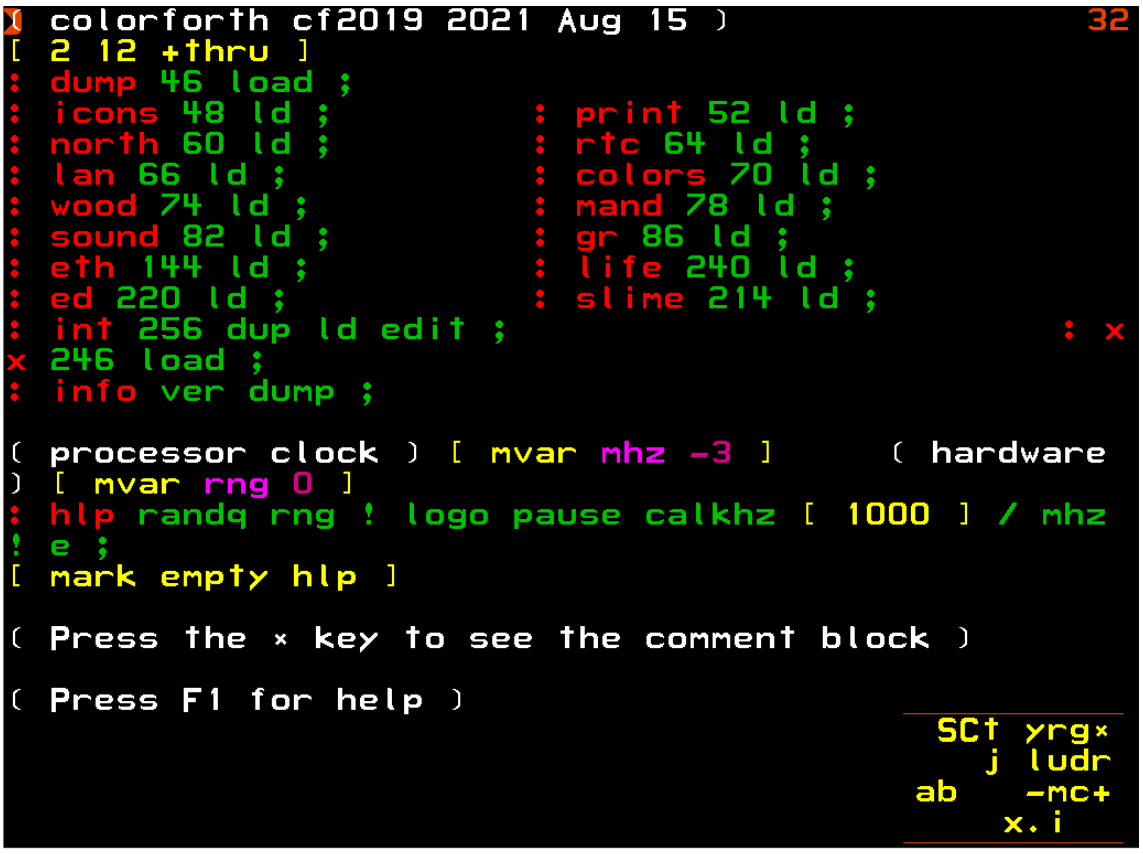


Note : you can press Alt-Enter on the keyboard to exit full screen mode, and from there you can click on the Power button to close Bochs.

The colorForth logo will flash briefly, then you should see block 32 in the colorForth editor :



Press the F4 key on the keyboard to select colour-blind mode :



Colour-blind mode adds ‘:’ to red words, ‘(‘ and ‘)’ around comments and ‘[‘ and ‘]’ around immediate words.

This makes the code look more like conventional Forth, and makes it usable by colour-blind people.

Press the space bar on the keyboard to exit the colorForth Editor :

```
( colorforth cf2019 2021 Aug 15 )                    32
[ 2 12 +thru ]
: dump 46 load ;
: icons 48 ld ;          : print 52 ld ;
: north 60 ld ;          : rtc 64 ld ;
: lan 66 ld ;            : colors 70 ld ;
: wood 74 ld ;           : mand 78 ld ;
: sound 82 ld ;          : gr 86 ld ;
: eth 144 ld ;           : life 240 ld ;
: ed 220 ld ;            : slime 214 ld ;
: int 256 dup ld edit ;                          : x
x 246 load ;
: info ver dump ;

( processor clock ) [ mvar mhz -3 ]      ( hardware
) [ mvar rng 0 ]
: hlp randq rng ! logo pause calkhz [ 1000 ] / mhz
! e ;
[ mark empty hlp ]

( Press the x key to see the comment block )

( Press F1 for help )
                                                pyfi gcrl
                                                aoeu htns
                                                qkxd bmwv
                                                     9x
```

Notice the yellow keypad in the bottom right-hand corner showing “pyfi gcrl” etc. Yellow means “immediate” – whatever you type will be executed immediately when you press the space bar. Note that the two orange lines above and below the keypad (that indicate editor mode) are no longer there.

From now on, refer to the **keypad** to select which character to type, not the **keyboard**. You can read cf2019_colorForth.pdf for a more detailed explanation.

*Figure 1 The **Keyboard***

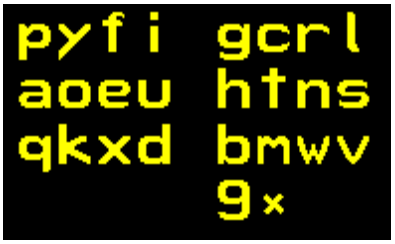


*Figure 2 The **Keypad***

The “9*” on the bottom row of the keypad mean that the Space bar will change to number mode, and the small ‘x’ means that the AltGr key will change to alternate characters (such as *). The asterisk character ‘*’ looks like a small ‘x’, because it represents multiplication. The ‘N’ key on the keyboard is often used to exit a mode.

The actions for each of the 27 keys of the keyboard that are used is hinted at by the corresponding character on the keypad.

There is no underscore character ‘_’, so I will use it in the instructions below to mean “press the space bar”.

Step 2 Edit block 250

Block 250 is where we will put our new program. Blocks 0 to 31 contain the primitive words of the colorForth system, blocks 32 to 64 contain the Forth words for the colorForth system, blocks 65 to 511 contain Apps.

_ \ press the space bar on the keyboard to enter number mode

Figure 3 The keypad in number mode

250 \ press the number keys to enter a number

This means, press the 'I' key, then the 'K' key on the keyboard, then whatever key is marked to the right of the 'L' key. On a German QWERTZ keyboard this is a 'Ö', on a QWERTY keyboard this will probably be a ';'.

Press the space bar again to go back to text mode – this is hinted at by the 'a' on the keypad :

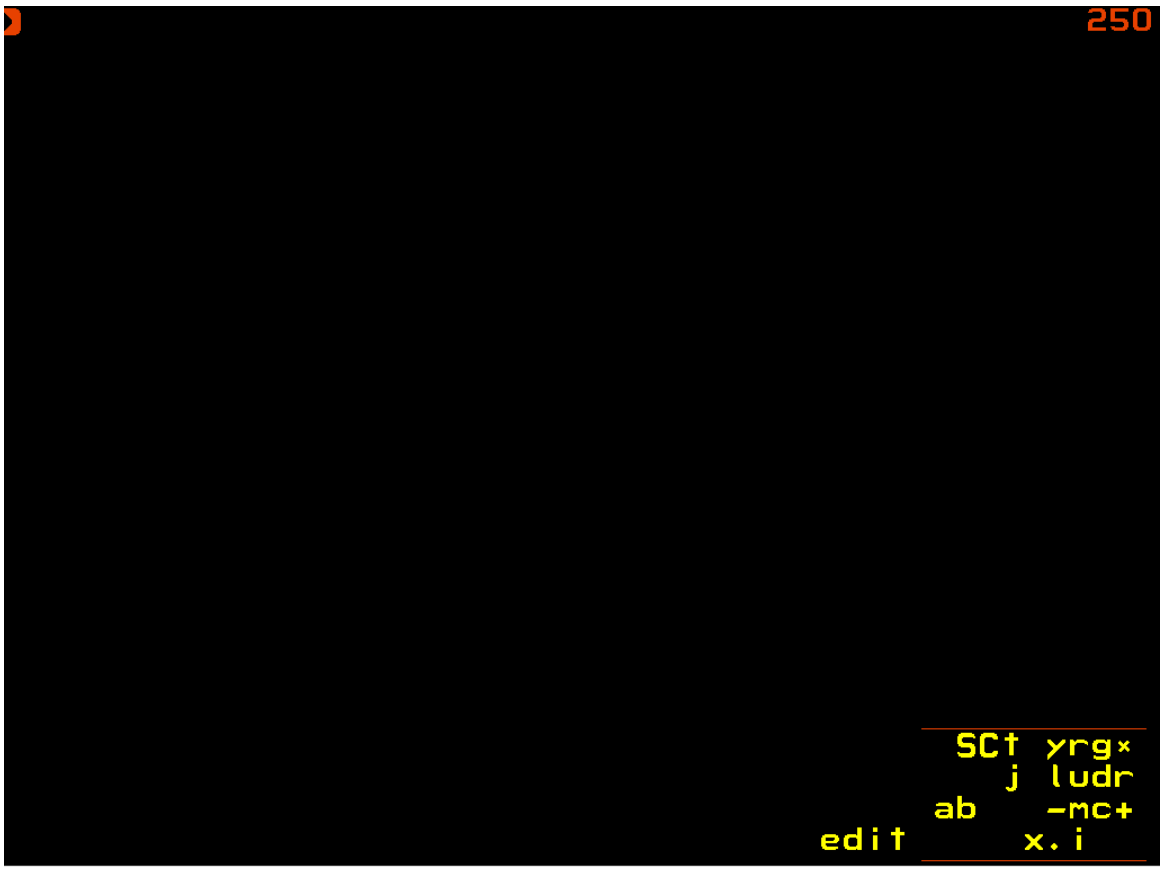
_ \ press the space bar on the keyboard to exit number mode

Now there is the number '250' on the stack :

edit_ \ type "edit" to edit block 250 (the '_' means the space bar)

To type "edit" you must press the keys DVRK on the keyboard, but it is best to try to forget about the keyboard and focus on the keypad. Note that the keypad is changing all the time.

The screen should now look like this :



Top left is the orange cursor that shows where the next Forth word will be inserted.

Top right is the block number being edited, 250 in this case. This block is currently empty.

Bottom left shows the stack, which is also empty at the moment (it used to look like this).

The orange lines above and below the keypad show that we are in edit mode.

The word “edit” to the left of the keypad shows the last Forth word that was entered. As each character of a word is typed the character’s value is accumulated on the stack, and the text representation of this accumulated value is shown to the left of the keypad.

If you have exited the editor and want to get back, type “e” in yellow immediate mode, (then the space bar) to edit the last block that was edited.

We are now editing block 250, using the main editor keypad :

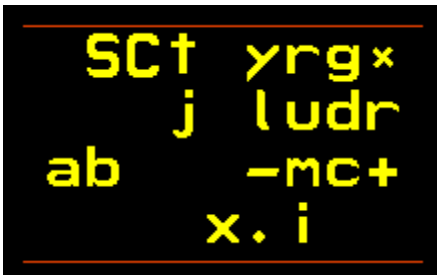


Figure 4 The main editor keypad

Step 3 Enter the program

To define a new Forth word we need to enter a red name, so press ‘r’ on the main editor keypad (that is the ‘I’ key on the keyboard). There is no need to type a colon (‘:’) because in colour-blind mode one is automatically added.

Figure 5 Creating the Forth word "fib", before pressing Space

Above shows the screen after having typed “fib”, but before pressing the space bar. The name “fib” is on top of the stack as a Huffman compressed value (46051), and is shown decompressed further to the right as “fib”.

Figure 6 Creating the Forth word "fib", after pressing Space

After pressing the space bar, the value on the stack (45061) has been removed and copied to the cursor position in the block. The editor displays the value in the first 32 bits of the block as the red text (with a ‘:’) “fib”.

The editor keypad has now gone green, indicating compiling mode. Note the ‘g’ in the bottom right of the keypad, indicating green (it was previously a red ‘r’).

The cursor has now moved until after the ": fib" text. There is no space character in the colorForth Huffman compressed 32 bit values, but one will be added automatically by the editor when the next word is displayed.

Now we can enter the green (compiled) code for the new Forth word **: fib push 1 dup pop -2 +**

It takes some practice to enter text and numbers using the keypad!

To enter the number '1' press the space bar without entering any other characters – this enters number mode.

Press the '1' using the keypad and then the space bar. To enter '-2' press the '-' on the keypad ('N' on the keyboard) then the number '2' ('I' on the keyboard). Then press the space bar.

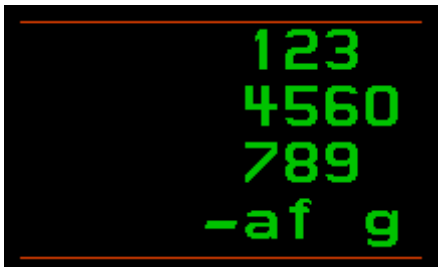
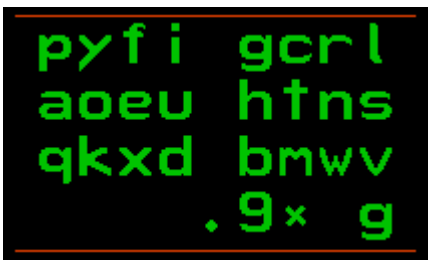


Figure 7 Number keypad

To type the '+' character, press '*' on the keypad (Alt-Gr on the keyboard) to enter alternate character mode :



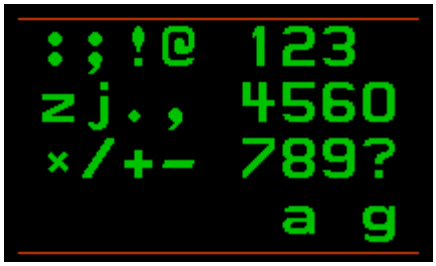


Figure 8 Alternate character keypad

Then press the '+' key on the keypad (the 'C' key on the keyboard) and then the space bar.

Step 4 Save your work

Having spent some time on entering part of the program, it is now a good time to save your work.

Exit the editor by pressing the '.' key on the keypad (the 'N' key on the keyboard), then press the space bar until you get back to the yellow immediate keypad :

Type "sa" on the keypad, then the space bar. The cf2019 system will be saved to the file cf2019.img.

Typing "sa" returns to the editor so you can add more words.

If you switch of Bochs and run **go.bat** again the cursor will have been reset to the start of the block. Use the "ludr"keys on the keypad, or the arrow keys on the keyboard, to move the cursor to after the '+".

Then you can press 'r' on the keypad to enter a new red definition.

: f -if drop nip ; then push swap over + pop -1 + f ;

Note that the '-' in "-if" is an alternate character, not a number, you must enter it using the alternate character keypad.

But the '-' in "-2" is a number and must be entered using the number keypad (press the '-' key before the '1').

Save your work again!

Step 5 Load and run the program

Exit the editor again by pressing the space bar, then type “ll” to load the current block.

“ll” is one of many double-letter words – press F1 on the keyboard to see a list.

Press the space bar to view immediate number keypad, press ‘8’ on the keypad (for example), then the space bar, then type “fib” to calculate the 8th Fibonacci number.

1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584....

You can see the number 8 on the stack, then the Huffman compressed word “fib”. Pressing the space bar gives the answer on the stack : **34**.

Note that I pressed F4 on the keyboard to leave colour-blind mode – the code fits better on one line.

Step 6 More features

You can add comments, stack comments and blue words (to format the text on the display) :

```
( App: Fibonacci )                                              250
: fib ( n-- ) push 1 dup pop -2 +
: f ( x...x ) -if drop nip ; cr
...     then push swap over + pop -1 + f ; cr
[ 8 fib ]
```

Press F3 to enable/disable the display of blue words. Press F4 to enable/disable colour-blind mode.

An alternative display looks like this :

```
App: Fibonacci                                                  250
fib n-- push 1 dup pop -2 +
f x...x -if drop nip ;
   then push swap over + pop -1 + f ;
8 fib
```

It is good practice to add a shadow block to explain what the App is doing :

```
Calculates the nth Fibonacci number                             251
```

Shadow blocks are in odd-numbered blocks, the Forth code is in even-numbered blocks, you can toggle between the two using the '*' key in the main editor keypad.

The yellow immediate words [8 fib] get executed when the block is loaded, so the answer 34 will get put on the stack. This saves some typing ;-)

The white comment on the top line "App: Fibonacci" marks this block as a cf2019 App.
Note that the first word must be exactly as written "App:" using the Capital 'C' mode in the main editor keypad.
Running the colorForth explorer App by typing "xx" shows all the Apps in the system :

```
              colorforth Explorer

70 Colors
74 Wood
78 Mandelbrot Set
82 Sounds make a noise
144 Ethernet
214 Slime : simple game
220 colorforth editor
240 Conways Game of Life
246 colorforth Explorer
250 Fibonacci fib n--
256 Timer Interrupt
258 Sounds
260 Test block :
262 Serial terminal
264 Mouse test
270 Floppy disk drive

                                                      e   ?

                                                      -   +
34 34                                  xx             .×
```

Use the '-' and '+' keys on the keypad to highlight an App, then press the space bar to run it. You can see that I ran the Fibonacci App twice, leaving two answers on the stack

Step 7 Save Your Work part 2

When you are happy with your program, you really must save it again in a different way.

Typing “sa” or similar words such as “ss” and “save” copies the code that you have written (which is in RAM) to the file **cf2019.img** on disk.

You must make a copy of **cf2019.img** as a backup, because running the file **goc.bat** will overwrite it and **delete all of your work.**

goc.bat compiles the system and then runs it, but is uses the refence copy of the system **cf2019Ref.img** as the source to create a new **cf2019.img**.

I like to use Windows to copy the file **cf2019.img** to the file **cf2019_2022Jan01.img** for example – a crude form of version control.

There is also a batch file **new.bat** that copies cf2019.img to cf2019Ref.img :

```
copy /b .\cf2019.img  .\cf2019Ref.img
```

Running **new.bat** will make your new program a part of the cf2019 system, so that running **goc.bat** will not delete it.

Note that running goc.bat restores the system to its “factory defaults” – you can use “xx” to find your App, or “250 edit” to view the code.