“Sally” Programming Language Demo

**Introduction:**

Sally is a simple concatenative programming language developed primarily for the purpose of gaining experience both with writing a lexical analyser and parser, and learning more about concatenative programming languages. Sally is primarily a toy language, i.e. I hope nobody would ever try to do anything important with it, but it has some pretty nifty features. Some of these can be seen in the examples.

To run the examples, it is first necessary to load them into Sally’s word dictionary. Simply enter the following at the Sally interactive mode command prompt:

“examples/examples” run

Notice that Sally does not know where to find the examples, hence the directory specification. The run word automatically appends the file extension (.sal) and loads, evaluates, and executes the contents of the examples file. Note that the examples file will also automatically load several other files, which contain other interesting examples to experiment with.

**Fibonacci Sequence:**

This example illustrates one of the iterative combinators which is built-in to Sally’s base package. The word fibonacci simply leaves the answer on the stack, while fibonacci-display will display the entire sequence up to the desired answer. Thus, type either something like:

5 fibonacci .s

To see the numbers 5 and 8 on the stack, or:

5 fibonacci-display

To see the numbers 1 1 2 3 5.

**Euclid’s GCD Algorithm:**

This example implements Euclid’s GCD algorithm using recursion. Type something like:

27 36 gcd print

To see the GCD of the two numbers (in this case, the answer will be 9).

**Sierpinski Gasket ASCII art:**

This is an extravagant example, but a simple line of code produces profound results. Enter the height and width of your console:

25 80 sierpinski

There is also an option with blocks (dependent on character set):

25 80 block-sierp

**Factorial Example:**

This example implements the factorial function recursively. Due to stack limits and such, this will fail for some size of n (may be system-dependent), but otherwise it is as simple as:

25 factorial

**File operations examples:**

These are used to demonstrate reading and writing files in Sally. As with everything else, the capabilities are extremely limited yet allow for a good deal of possibilities within these limitations. Simply push a filespec to the stack and call:

“examples/test.bin” fwrite-ex

“examples/test.bin” fread-ex

Note that the writing example will overwrite anything already present in the passed filespec, so don’t pass it anything important! The writing example shows how Sally can write a binary-correct file, while the reading example will simply read in a file and display it on the screen. As with everything else in Sally, data is pushed to the stack, so the file must be of an appropriate length that it may be stored in a Python string and/or on the stack.

**Other example:**

This is best left as a surprise ;) Type the following in and see what happens:

“examples/brainf2” run “examples/hw.bf” bfexec