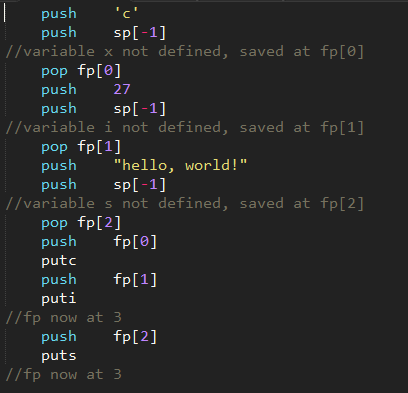
Compiler Project

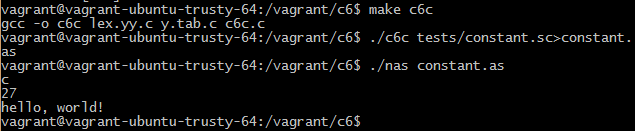
# Constants

integers, chars and immutable strings.

test file:constant.sc

assembly file:constant.as





Above is a screenshot of the code running.

# Variables

variables are alphanumeric, up to 12 chars long and have no particular type.

test file: var.sc

assembly file:var.as



variables can be either local or global. You can refer to a global variable inside a function with “@VAR”

test file:var2.sc

assembly file:var2.as



# Arrays

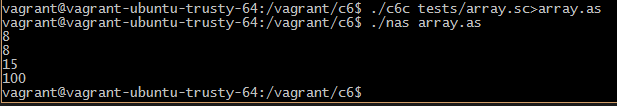
We support multidimensional arrays.

Arrays of integers and chars can be initialized with the same value.

The syntax for declaring a multidimensional array is: array matrix[5,4,3]

test file:array.sc

assembly file:array.as



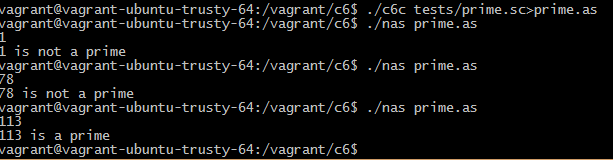
Arrays of chars can also be initialized by a string. (so that the array is a real string)

# Functions

Functions accept none or multiple arguments. They must appear before they are used and they must return with a value

test file:prime.sc(this is a function that tests whether the input number is a prime)

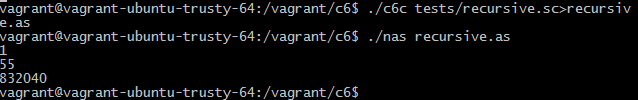
assembly file:prime.as



Another test for recursive functions:

test file:recursive.sc(a function that calculates the Fibonacci sequence numbers recursively)

assemble file:recursive.as



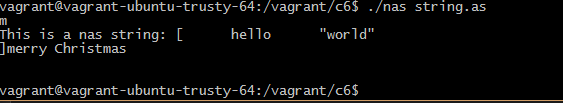
# Real Strings

We can initialize a char array with a string, or output a real string.

We also support escape characters like \n \\ \t.

test file:string.sc

assemble file:string.as



# Input, Output

input syntax: read, geti,getc,gets (read is the same as geti for a single variable)

output syntax:print, puti,putc,puts(print is the same as puti for a single variable)

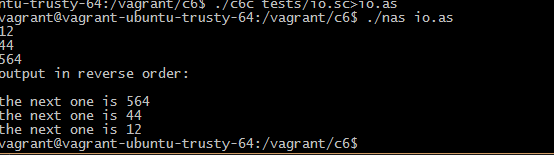
We support multiple gets and multiple puts.

We also support printing with a format string for the basic types (not including arrays).We assume that if the first argument is a string, it is the format string.

We can also print out integer arrays. The numbers will be printed out one by one without spaces in between.

test file:io.sc

assembly file:io.as

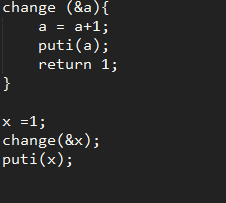


# Pass by Reference

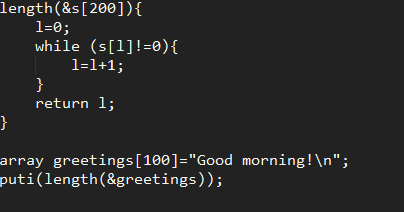
Function parameters can be passed by reference.

‘&’ needs to be used in the function definition and the function call.

test files: ref.sc ref2.sc strlen.sc









# Application - sortplot

A very simple application that repeatedly receives an integer as input and output the sorted list of all integers and draw them in a line graph (with matplotlib.pyplot and numpy in Python3).

Usage (working directory: tests/sortplot/, requires matplotlib, numpy and python3):

../../nas sortplot.as | python3 receiver.py

input any number except -1: draw the sorted data since startup

input -1: exit

