NORG

Introduction

This esolang originally started out as a small homage to the most beloved species ☺ in the whole galaxy. It however adapted somewhat to humans so I felt it needed a different name. NORG has the following characteristics:

Each program starts with creating a cube of nxnxn integer cells and puts the current position (cursor) at the center. All cells have a (signed) value register (8 Bytes), and an execution register (width 20 Bytes). These are all initially set to 0 and empty resp.. There are also some global flags:

- a comparison indicator (initially 0)
- a file status indicator (initially 0)

In addition there are three global value registers (initially 0), see below.

Each command is given as a sequence of 1-n characters, the first being the command and the others the parameters (if any). Commands cannot be separated by unused characters - except for newlines.

The language only recognizes the characters given in the translation table below. On reading a program file, any unrecognized character causes the program to end with error code 1 - with the exception of newline characters: these are ignored. So you may separate different parts of the program by using them.

Commands

The first char of a program has a special meaning: it defines the length of the cube, so the length may be 1..39 – see translation table. (You should not set it to 0.)

command	param	Meaning	Explanation
a	b or l	assimilate	if no file is selected: select the first file in the same directory that does not have the extension .norg, then proceed with the selected file. ab = read the next byte and set the cell to its value (see translation table) al = read the next line assume it contains a number and set the cell to its value if end of file is reached then delete the file (a)
b	none	move back	move the cursor backward
c	(command)	condition	next command is only executed if comparison indicator is 1
d	none	move down	move the cursor down
e	none or digit	execute	e=execute the current cell exec. register content (see command t). e<n>=where n is digit: exec the nth defined exec register regardless of the current pos. (en) Both is done by inserting the code into the code string right after the current position.
f	none	move front	move the cursor forward
g	direction	get	get the value from cell in direction given by the parameter and overwrite current cell.
h	direction	get exec reg	get the exec register content from cell in direction given and overwrite current cell
i	i, v or q	invert	ii = invert global comparison indicator iv = invert current cell value

			iq = invert current global register
j	none	jump	end current e command sequence
l	none	move left	move the cursor left
m	none	move	mm = move cursor mode: wrap (default) ↔ page mode, see below mo = set cursor to origin (0,0,0) mc = set cursor to center
o	none	output mode	change output mode: numeric (initial) → alphanumeric → alpha only → numeric
q	direction or p,n,g	global registers	3 global registers exist: 0,1, and 2. Initially g.r. 0 is selected. Selection can be changed by: qp = select previous register qn = select next register other commands: qg=set selected g.r. to the current cell value qr/l/b/f/u/d: same as sr/l/b/f/u/d but the g.r. value is sent to the current cell, triggering the action associated with the direction
r	none	move right	move the cursor right
s	direction	send	send the content of the current cell to the cell in the direction given as parameter, see below
t	commands	set exec register	the t command may be followed by up to 20 command chars and is terminated by a dot ‘.‘ The register is set to the command sequence given as params to t. Previous contents of the register are deleted. (t)
u	none	move up	move the cursor up
x	none	exit	program ends.
y	s,o,p or digit	output	yp=outputs the current plane as seen from the front y<n>, n=digit: outputs the plane no. n as seen from the front ys = outputs the current value + newline yo = outputs the current value only
z	(com-	file stat con-	next command is only executed if file status

	mand)	dition	indicator is 1
others	none	set	if the char is recognized: set the cell content to the value of the char, otherwise program stops

(a)
if there are no more files to assimilate, the file status indicator is set to 1
(en)
the index is determined by the sequence of t commands in the original command string. Note that e<n> accesses the code currently saved in the cell associated with the <n>th t-command, so subsequent changes to the cell code will lead to a different executions sequence.
(t)
The max. length of a command string is 20 Bytes. Note that leading zeroes are ignored.

direction = one of d(own), u(p),r(ight),l(eft),f(ront),b(ack)

sending the content of a cell results in the following action:

- direction right: current cell content is added to right cell
- direction left: current cell content is set to the minimum of its value and the left cell value
- direction up: top cell content is multiplied with current cell content
- direction down: down cell content is divided by current cell content
- direction front: global comparison indicator is set to 1 if both cell contents are equal, -1 otherwise
- direction back: global comparison indicator is set to1 if back cell content >= front cell content, -1 otherwise

cursor mode

If a non-existing cell is referenced by one of the commands r,l,u,d,f,b,s,g the cursor is set in the following way:

in wrap mode (default):

- direction right: referenced cell is the leftmost cell in the same line
- direction left: referenced cell is the rightmost cell in the same line
- direction up: referenced cell is the bottom cell in the same line
- direction down: referenced cell is the topmost cell in the same line
- direction front: referenced cell is the backmost cell in the same line
- direction back: referenced cell is the frontmost cell in the same line

in page mode:
the cursor first wraps in the same way as in wrap mode and then additionally moves as follows (which may trigger another page mode movement):

- direction right: move down
- direction left: move up
- direction up: move front
- direction down: move back
- direction front: error, stop program
- direction back: error, stop program

Notes

- Since the number of (scalar) variables is confined to at most 79*79*79 = 493,039, the language – strictly speaking – isn’t turing complete, see definition of WHILE language…
- The language doesn’t support comments in any way because these are irrelevant ☺. Well, the code looks linear but the underlying structure is cubic. So you should better document your programs using some kind of spreadsheet.
  If you want to include your name as the author of the program, you may achieve this by using the t-command like so:
  t Author is .t John Doe .t0.
- There is no way to use standard input to supply data to a program, you have to use the ‘a‘ command for reading files. But be aware of the fact that files are deleted if eof is reached…
- Booleans: the comparison indicator can be seen as a boolean value: 1 = true, -1 = false, 0 = undecided. It can be set by sb and sf. The only way to really use it is the ‘c‘ command.

Translation table

Note that integers do not correspond to ascii characters but are coded like this:

int	**char**
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	a
11	b
..	
35	z
36	A
37	B
..	
61	Z
62	.
63	,
64	!
65	?
66	:
67	;
68	-
69	„=“
70	<

71	>
72	+
73	*
74	/
75	“
76	‘
77	{
78	}
79	space
-1	all other chars