Lecture 1: Malbolge, A Quick Guide

Malbolge, named after the eighth circle of Hell in Dante's *Inferno*, is a language specifically designed to be impossible to write useful programs in. It took two years before anyone discovered how to write "Hello, World[1]" in it: furthermore, this wasn't even done by humans. (A beam-search algorithm[2] was used to generate the program.)

You may have noticed that Nic and Asilata's class "5 Programming Languages in 10 Days" *somehow* forgot to talk about this language! Let's fix that.

1 Malbolge: Setup and Instruction Sets

Registers. Malbolge has three registers, a, c, and d. When the program starts, all three of these registers are 0; as the program runs[3], these values may change. The register c is special; it points to the current **instruction**. d and a, conversely, are usually somehow related to whatever data you're currently manipulating.

Memory. Binary is boring. Accordingly, Malbolge works in ternary! Specifically, Malbolge runs in a block of $3^{10} = 59048$ memory locations, each of which contains a number of length 3^{10}. Conveniently, this allows any one of these blocks x to either be interpreted as a number (in which case we write it as x) or as an address, pointing to the value stored in one of our other 3^{10} blocks (in which case we write it as $[x]$.)

Instructions. Malbolge has eight instructions. To figure out what instruction to do at any point in time, Malbolge does the most natural thing possible: it takes the value at $[c]$, adds the number c to it, and takes that sum mod 94. After doing this, perform the corresponding operation:

[1] Or, more accurately, "HEllO WORld".

[2] Roughly speaking, this search takes a program, generates a number of possible "successors" to it by adding random little bits to the end of them, picks a handful that it thinks are likely to work out at the end because they're at least printing out something, and then repeats this search on the successors. Basically a miniaturized version of evolution.

[3] Assuming that your program does run.

Value of $([c]+c) \mod 94$	Instruction	Result
4	$jump(d)$	Set c, the code pointer, to the value at $[d]$.
5	$print(a)$	Print the character given by a, mod 256, as an ASCII character.
23	$a = input$	Take a character from standard input, put it in a.
39	$a = [d] = rotate([d])$	Take the ternary string at $[d]$, rotate it around to the right, put it in a and $[d]$.
40	$d = [d]$	Put the value at $[d]$ into d.
62	$a = [d] = crazy([d], a)$	Perform the crazy operation using the value at $[d]$ and a, and store the result at $[d], a$.
68	nop	Does nothing.
81	$halt$	Halts.

The crazy operation referenced above is a trit-wise operation on two ternary strings of length k that returns a ternary string of length k. It works character-by-character on the ternary string by using the following table:

	0	1	2
0	1	0	0
1	1	0	2
2	2	2	1

After each instruction, Malbolge helpfully takes the value at $[c]$, replaces it with itself mod 94, and then encrypts the result using the following table:

result	encrypted	**result**	encrypted	**result**	encrypted	**result**	encrypted	**result**	encrypted
0	57	19	108	38	113	57	91	76	79
1	109	20	125	39	116	58	37	77	65
2	60	21	82	40	121	59	92	78	49
3	46	22	69	41	102	60	51	79	67
4	84	23	111	42	114	61	100	80	66
5	86	24	107	43	36	62	76	81	54
6	97	25	78	44	40	63	43	82	118
7	99	26	58	45	119	64	81	83	94
8	96	27	35	46	101	65	59	84	61
9	117	28	63	47	52	66	62	85	73
10	89	29	71	48	123	67	85	86	95
11	42	30	34	49	87	68	33	87	48
12	77	31	105	50	80	69	112	88	47
13	75	32	64	51	41	70	74	89	56
14	39	33	53	52	72	71	83	90	124
15	88	34	122	53	45	72	55	91	106
16	126	35	93	54	90	73	50	92	115
17	120	36	38	55	110	74	70	93	98
18	68	37	103	56	44	75	104		

This step stops you from accidentally repeating any given instructions, so that your code is always new and interesting!

Finally, once you've done this step, you increase both c and d by 1, and repeat the execution cycle. A html compiler can be found at

http://matthias-ernst.eu/malbolge/debugger.html

You enter your code via ASCII values (i.e. each ascii character is a ternary number.) It doesn't support input yet, but I'm sure you can still do fascinating things without it.