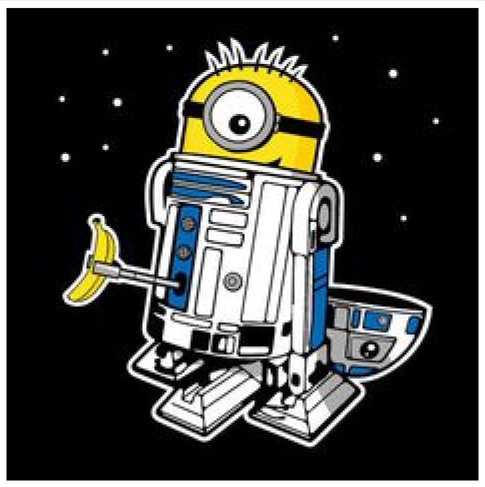
R2M2

RADARE2 + MIASM2 = LOVE

SSTIC2016 - @guedou

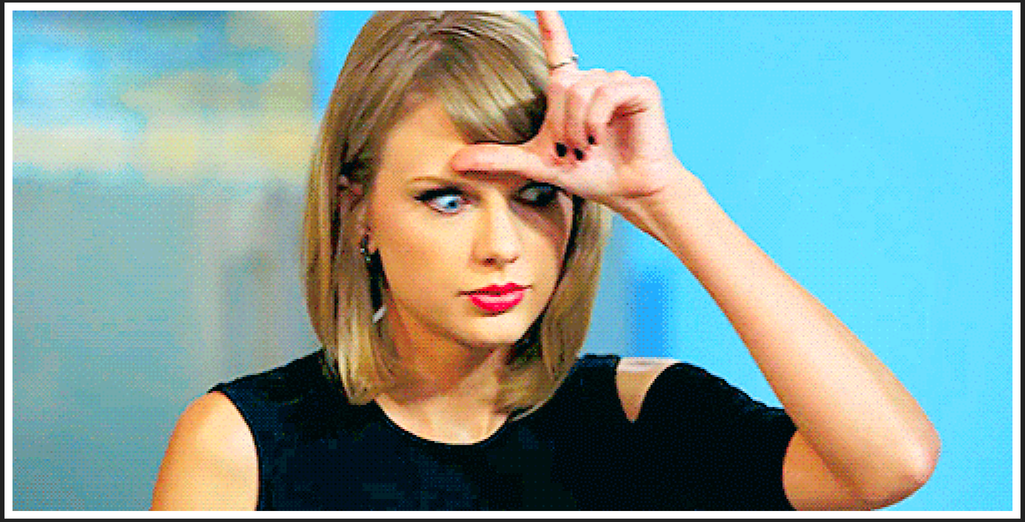


GOALS?

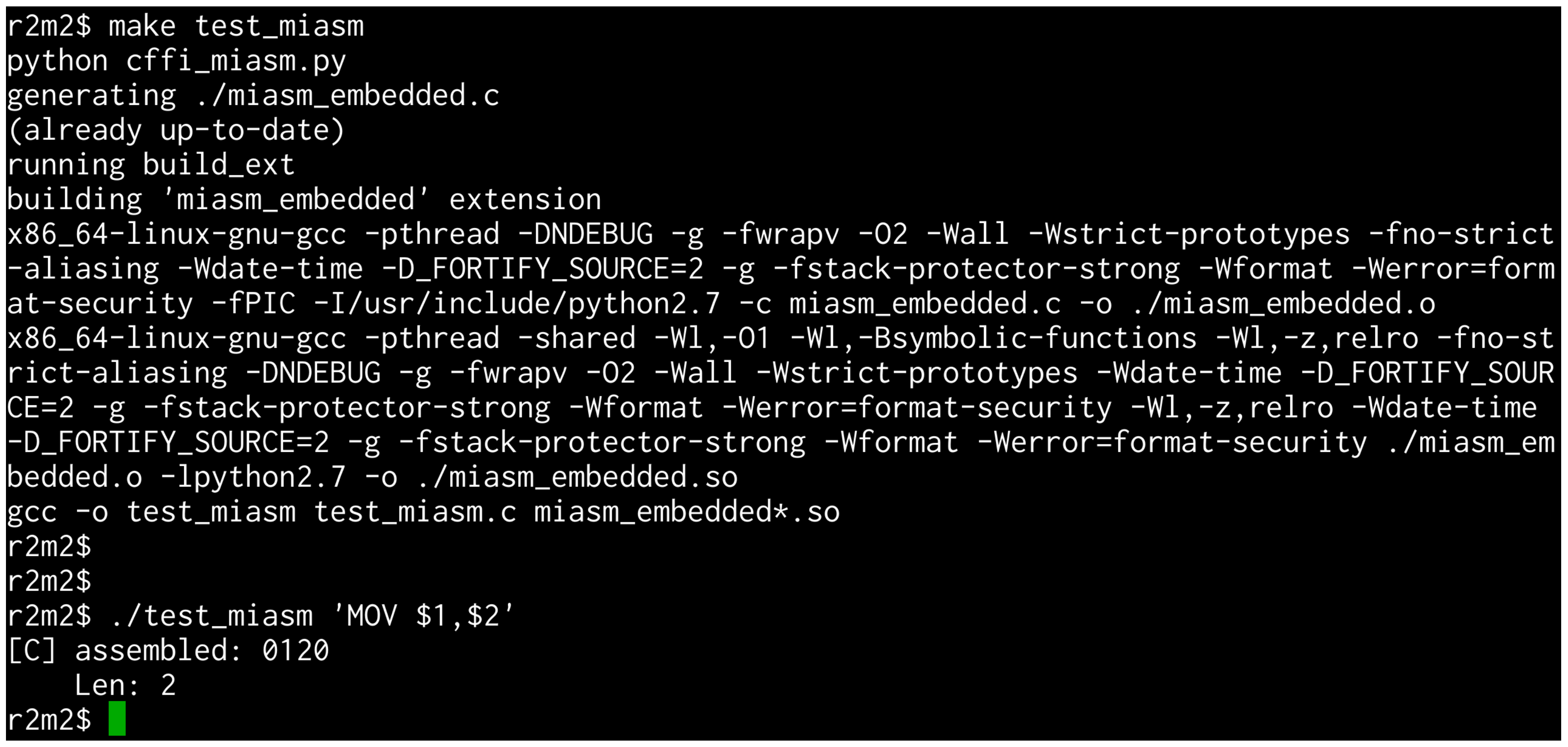
r2m2 is a radare2 plugin that aims to:

- use radare2 as a frontend to miasm2
  - tools, GUI, shortcuts, ...
- use miasm2 as a backend to radare2
  - asm/disas engine, symbolic execution, ...

MER IL ES FOU ?!?



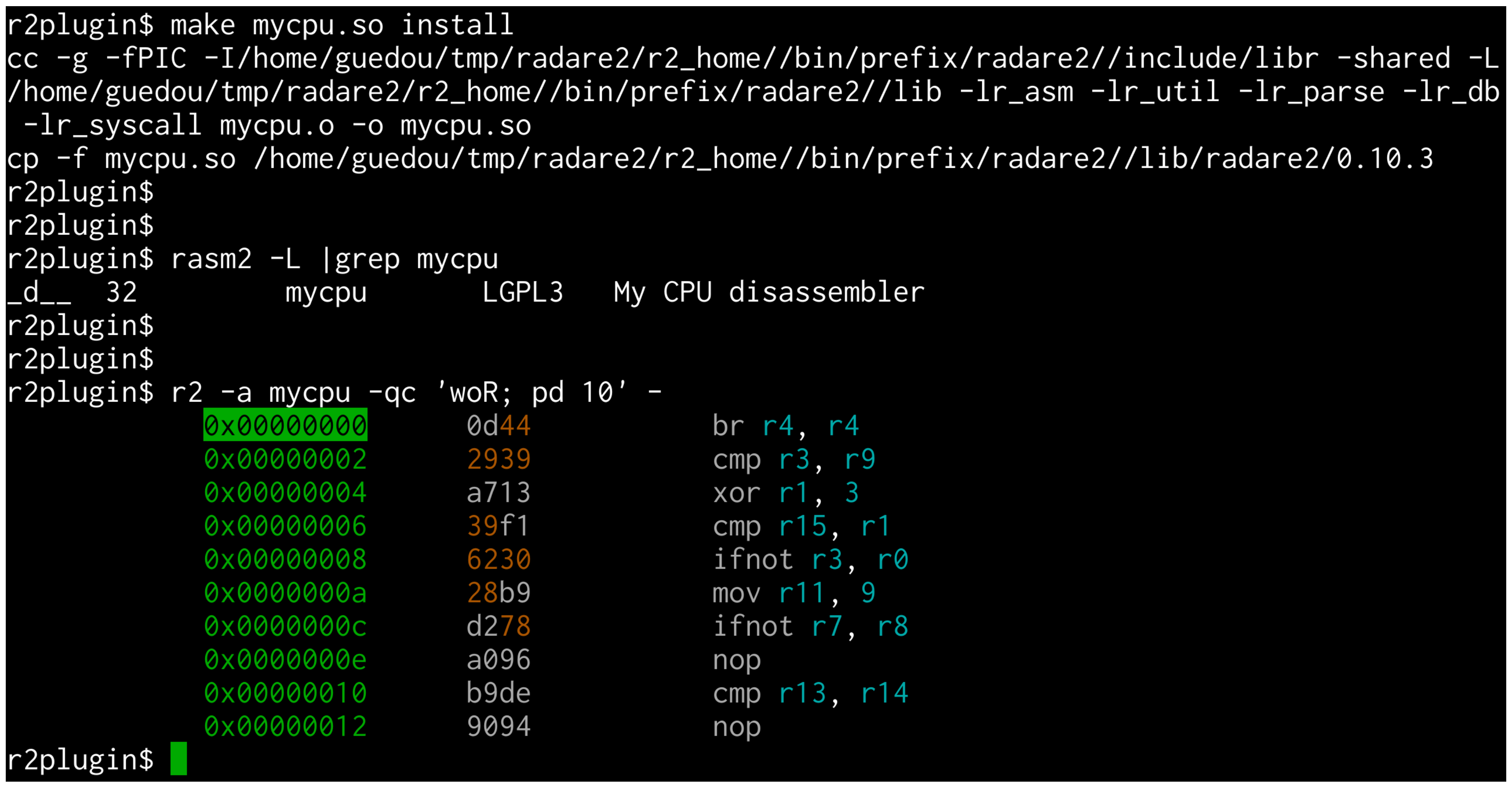
STEP #1 - CALL PYTHON FROM C



```
r2m2$ make test_miasm
python cffi_miasm.py
generating ./miasm_embedded.c
(already up-to-date)
running build_ext
building 'miasm_embedded' extension
x86_64-linux-gnu-gcc -pthread -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototypes -fno-strict
-aliasing -Wdate-time -D_FORTIFY_SOURCE=2 -g -fstack-protector-strong -Wformat -Werror=form
at-security -fPIC -I/usr/include/python2.7 -c miasm_embedded.c -o ./miasm_embedded.o
x86_64-linux-gnu-gcc -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -Wl,-z,relro -fno-st
rict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall -Wstrict-prototypes -Wdate-time -D_FORTIFY_SOUR
CE=2 -g -fstack-protector-strong -Wformat -Werror=format-security -Wl,-z,relro -Wdate-time
-D_FORTIFY_SOURCE=2 -g -fstack-protector-strong -Wformat -Werror=format-security ./miasm_em
bedded.o -lpython2.7 -o ./miasm_embedded.so
gcc -o test_miasm test_miasm.c miasm_embedded*.so
r2m2$
r2m2$
r2m2$ ./test_miasm 'MOV $1,$2'
[C] assembled: 0120
    Len: 2
r2m2$ █
```

The cffi Python module produces a `.so`

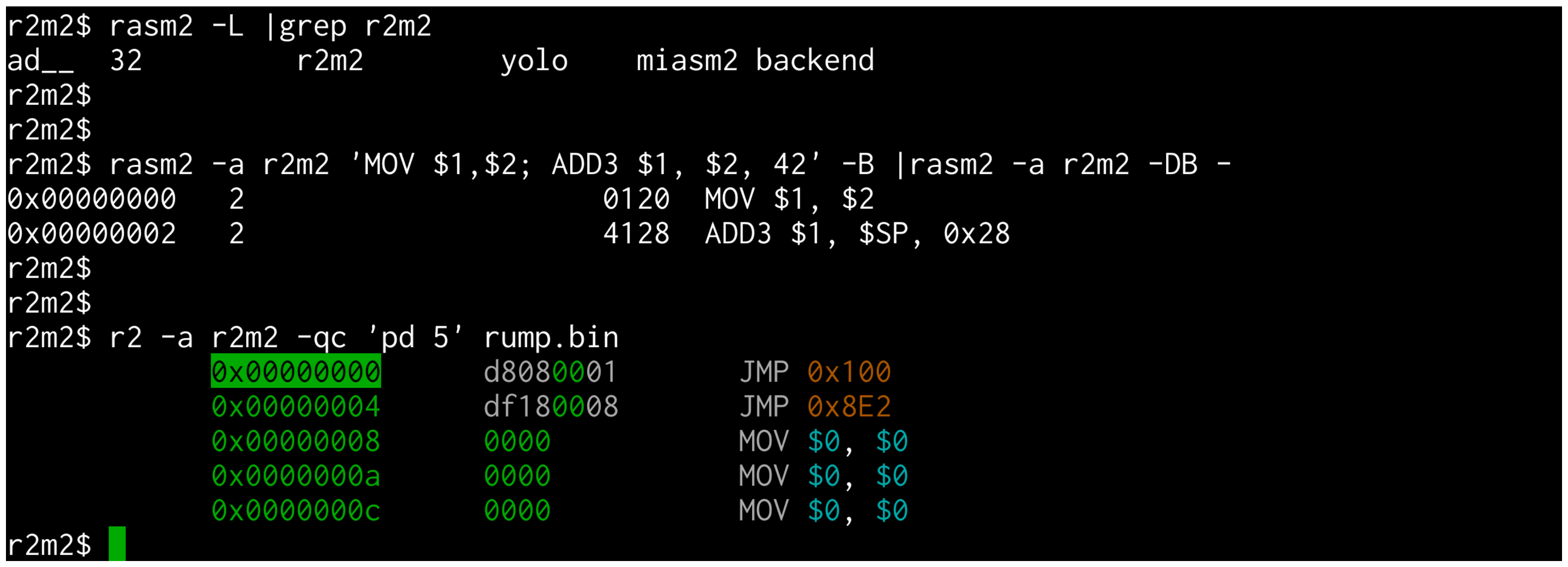
STEP #2 - BUILD A RADARE2 PLUGIN



```
r2plugin$ make mycpu.so install
cc -g -fPIC -I/home/guedou/tmp/radare2/r2_home//bin/prefix/radare2//include/libr -shared -L
/home/guedou/tmp/radare2/r2_home//bin/prefix/radare2//lib -lr_asm -lr_util -lr_parse -lr_db
 -lr_syscall mycpu.o -o mycpu.so
cp -f mycpu.so /home/guedou/tmp/radare2/r2_home//bin/prefix/radare2//lib/radare2/0.10.3
r2plugin$
r2plugin$
r2plugin$ rasm2 -L |grep mycpu
_d__  32         mycpu      LGPL3   My CPU disassembler
r2plugin$
r2plugin$
r2plugin$ r2 -a mycpu -qc 'woR; pd 10' -
            0x00000000      0d44           br r4, r4
            0x00000002      2939           cmp r3, r9
            0x00000004      a713           xor r1, 3
            0x00000006      39f1           cmp r15, r1
            0x00000008      6230           ifnot r3, r0
            0x0000000a      28b9           mov r11, 9
            0x0000000c      d278           ifnot r7, r8
            0x0000000e      a096           nop
            0x00000010      b9de           cmp r13, r14
            0x00000012      9094           nop
r2plugin$
```

The r2 Wiki shows on to add a new architecture

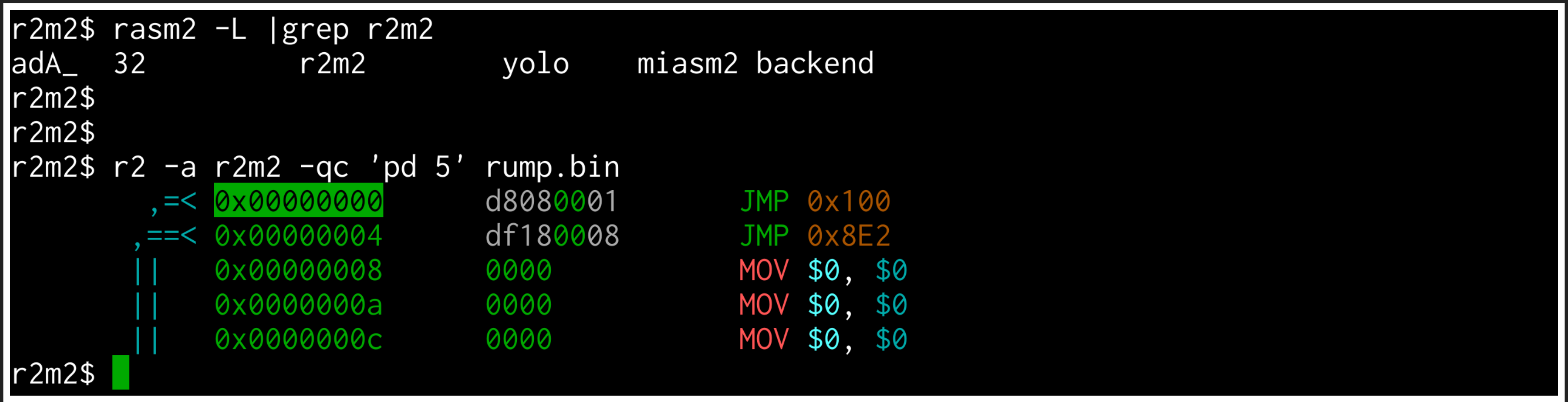
STEP #3 - SHAKE WELL



```
r2m2$ rasm2 -L |grep r2m2
ad__  32         r2m2        yolo    miasm2 backend
r2m2$
r2m2$
r2m2$ rasm2 -a r2m2 'MOV $1,$2; ADD3 $1, $2, 42' -B |rasm2 -a r2m2 -DB -
0x00000000   2                     0120  MOV $1, $2
0x00000002   2                     4128  ADD3 $1, $SP, 0x28
r2m2$
r2m2$
r2m2$ r2 -a r2m2 -qc 'pd 5' rump.bin
            0x00000000      d8080001       JMP 0x100
            0x00000004      df180008       JMP 0x8E2
            0x00000008      0000           MOV $0, $0
            0x0000000a      0000           MOV $0, $0
            0x0000000c      0000           MOV $0, $0
r2m2$
```

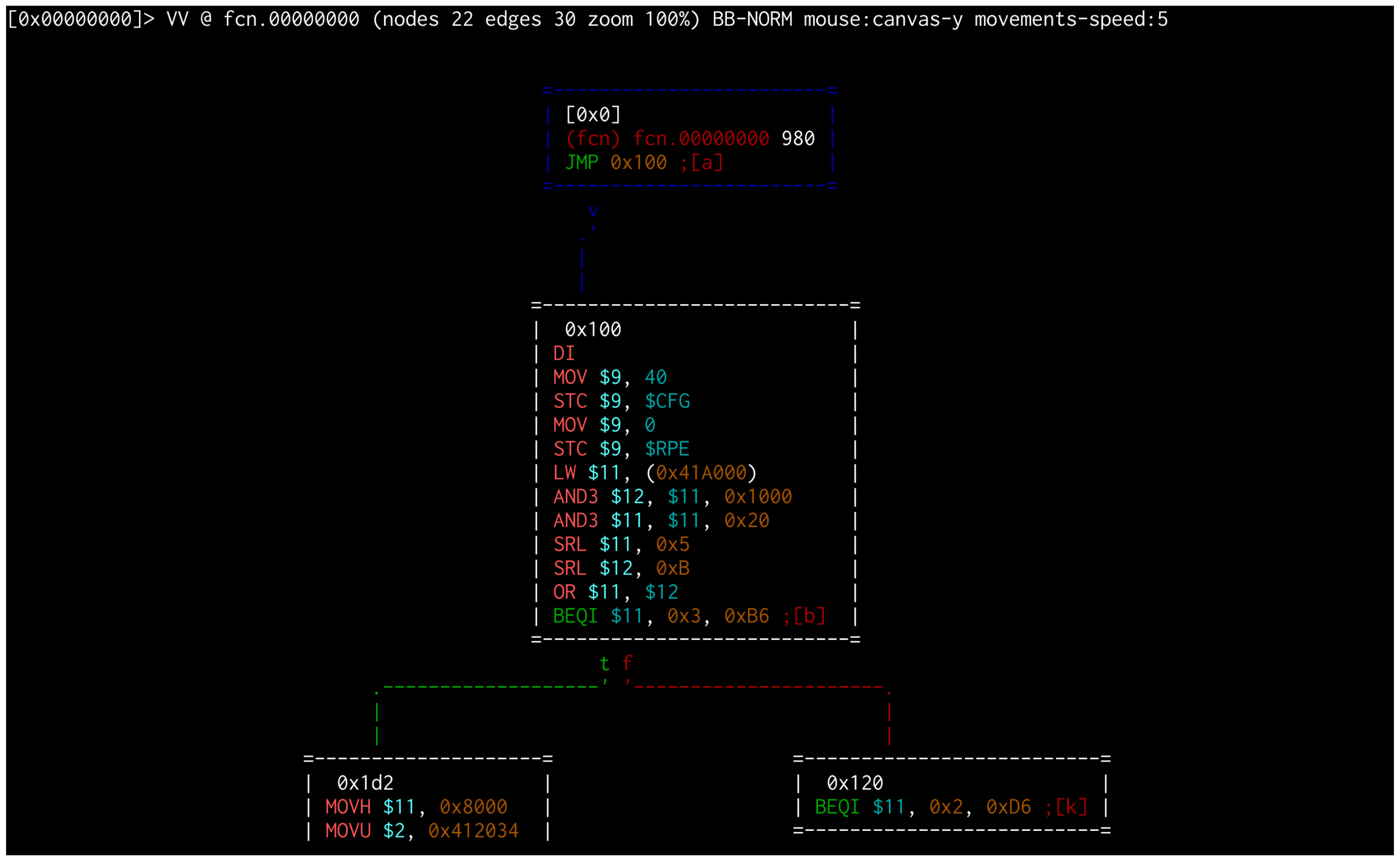
assemble() & disassemble() must be implemented

STEP #4 - CALL GRAPH



```
r2m2$ rasm2 -L |grep r2m2
adA_  32         r2m2        yolo    miasm2 backend
r2m2$
r2m2$
r2m2$ r2 -a r2m2 -qc 'pd 5' rump.bin
        ,=< 0x00000000      d8080001       JMP 0x100
       ,==< 0x00000004      df180008       JMP 0x8E2
       ||   0x00000008      0000           MOV $0, $0
       ||   0x0000000a      0000           MOV $0, $0
       ||   0x0000000c      0000           MOV $0, $0
r2m2$
```

Use miasm2 to *classify* opcodes according to radare2 types

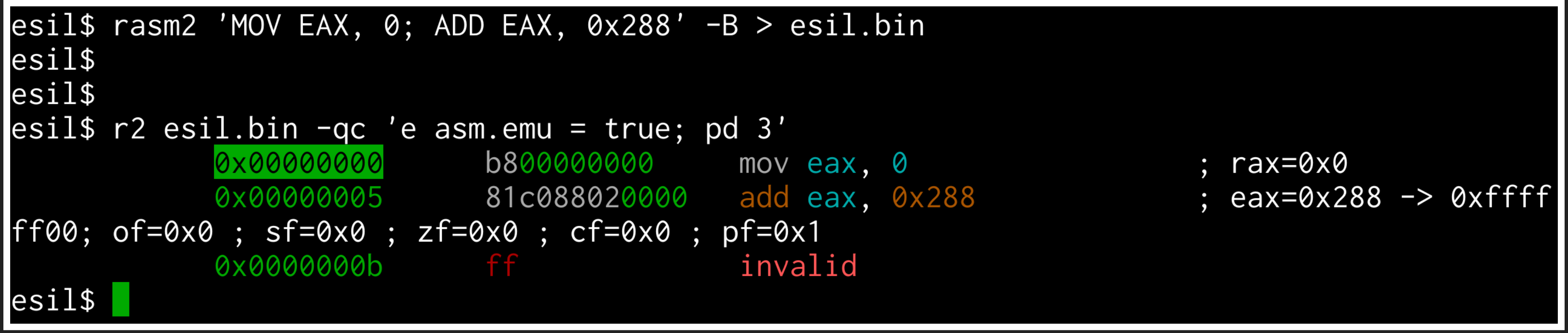


```
[0x00000000]> VV @ fcn.00000000 (nodes 22 edges 30 zoom 100%) BB-NORM mouse:canvas-y movements-speed:5


                                               =------------------------=
                                               | [0x0]                  |
                                               | (fcn) fcn.00000000 980 |
                                               | JMP 0x100 ;[a]         |
                                               =------------------------=
                                                   v
                                                   '
                                                  .
                                                  |
                                                  |
                                              =---------------------------=
                                              |  0x100                    |
                                              | DI                        |
                                              | MOV $9, 40                |
                                              | STC $9, $CFG              |
                                              | MOV $9, 0                 |
                                              | STC $9, $RPE              |
                                              | LW $11, (0x41A000)        |
                                              | AND3 $12, $11, 0x1000     |
                                              | AND3 $11, $11, 0x20       |
                                              | SRL $11, 0x5              |
                                              | SRL $12, 0xB              |
                                              | OR $11, $12               |
                                              | BEQI $11, 0x3, 0xB6 ;[b]  |
                                              =---------------------------=
                                                    t f
                                .-------------------' '----------------------.
                                |                                            |
                                |                                            |
                          =---------------------=                    =---------------------------=
                          |  0x1d2              |                    |  0x120                    |
                          | MOVH $11, 0x8000    |                    | BEQI $11, 0x2, 0xD6 ;[k]  |
                          | MOVU $2, 0x412034   |                    =---------------------------=
```

NEXT STEPS?

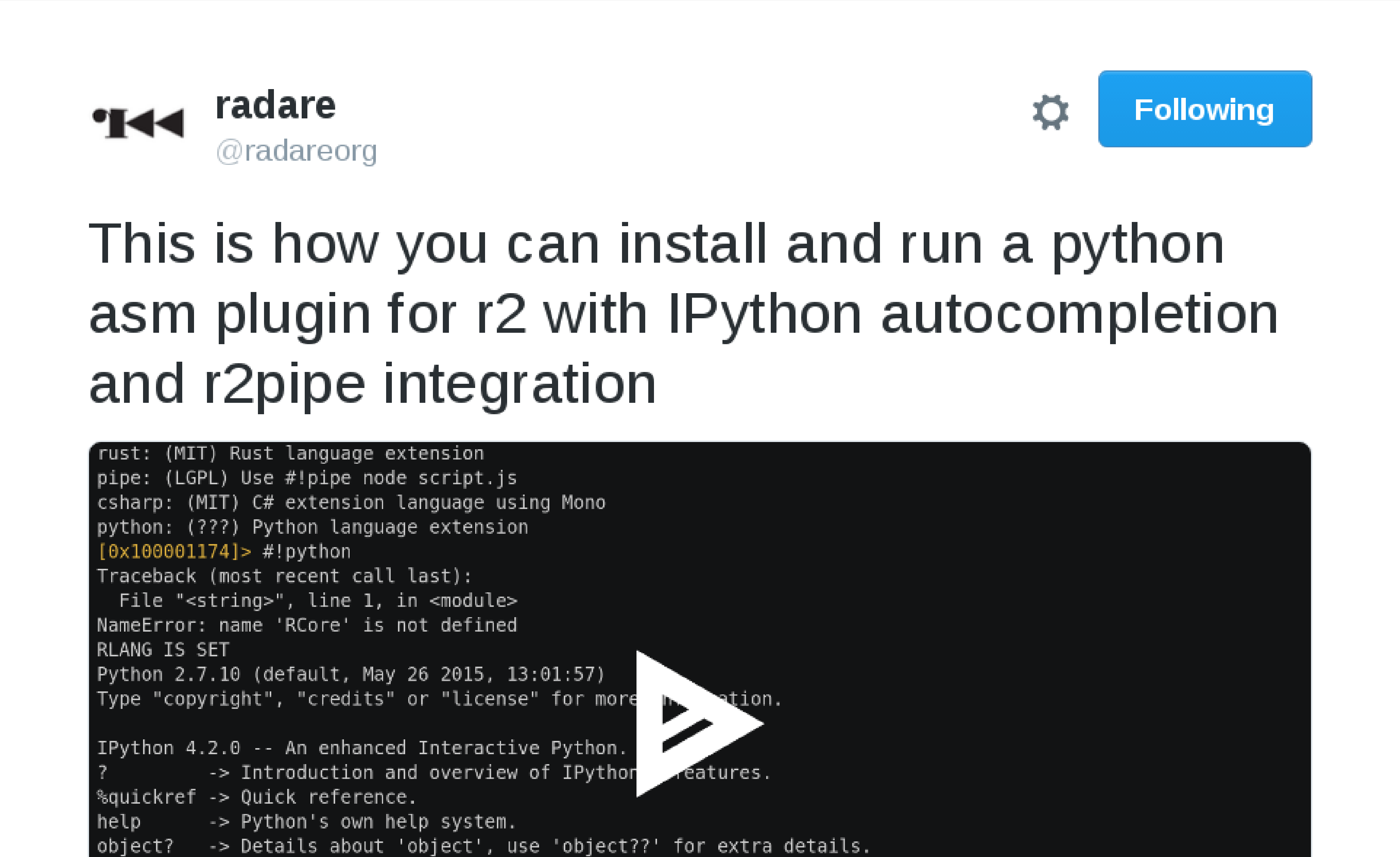
1/ Convert m2 expressions to r2 esil



```
esil$ rasm2 'MOV EAX, 0; ADD EAX, 0x288' -B > esil.bin
esil$
esil$
esil$ r2 esil.bin -qc 'e asm.emu = true; pd 3'
            0x00000000      b800000000     mov eax, 0                 ; rax=0x0
            0x00000005      81c088020000   add eax, 0x288             ; eax=0x288 -> 0xffff
ff00; of=0x0 ; sf=0x0 ; zf=0x0 ; cf=0x0 ; pf=0x1
            0x0000000b      ff             invalid
esil$ █
```

2/ Use the radare2 plugin API

See video & code



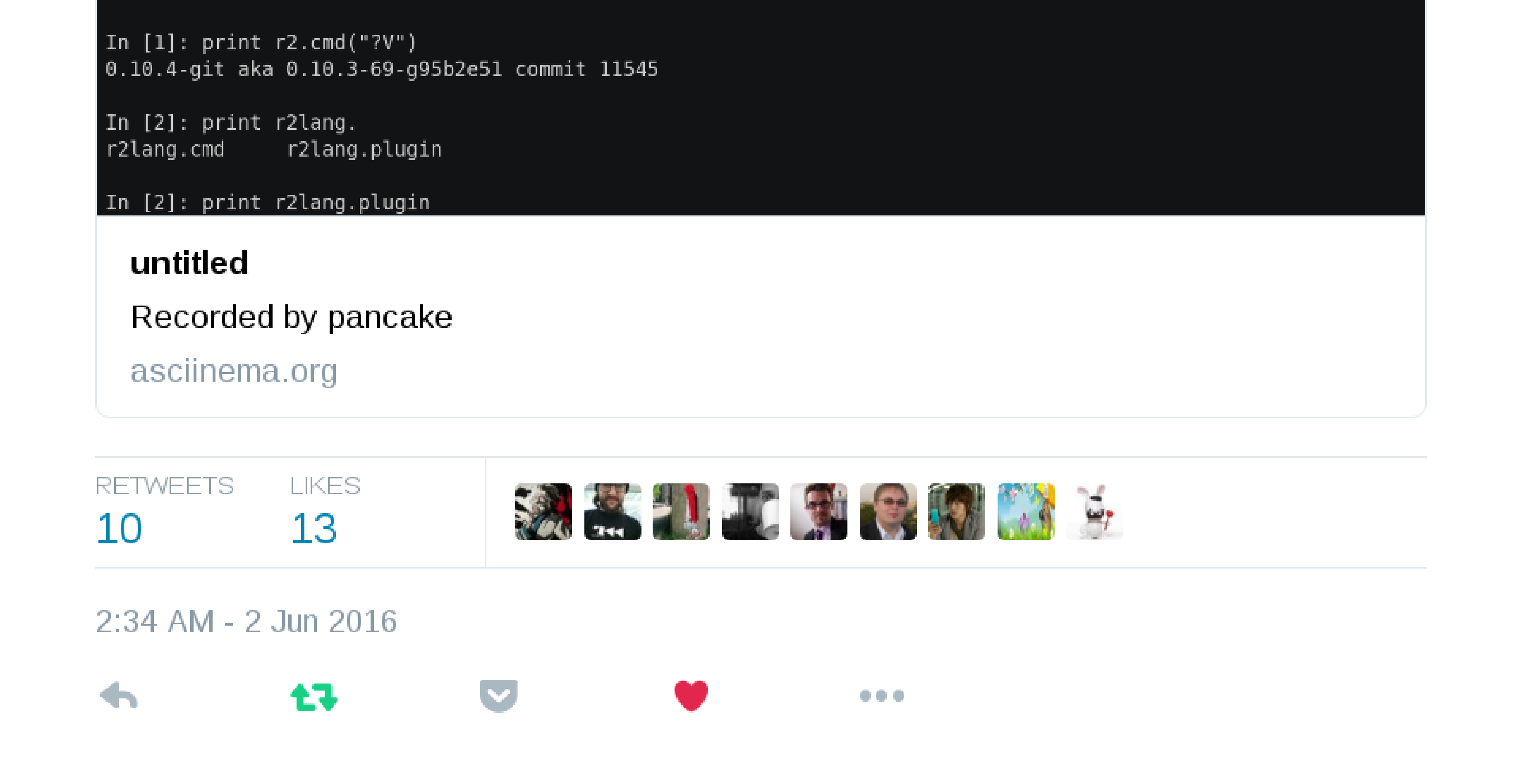
radare
@radareorg

Following

This is how you can install and run a python asm plugin for r2 with IPython autocompletion and r2pipe integration

```
rust: (MIT) Rust language extension
pipe: (LGPL) Use #!pipe node script.js
csharp: (MIT) C# extension language using Mono
python: (???) Python language extension
[0x100001174]> #!python
Traceback (most recent call last):
  File "<string>", line 1, in <module>
NameError: name 'RCore' is not defined
RLANG IS SET
Python 2.7.10 (default, May 26 2015, 13:01:57)
Type "copyright", "credits" or "license" for more information.

IPython 4.2.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
```



```
In [1]: print r2.cmd("?V")
0.10.4-git aka 0.10.3-69-g95b2e51 commit 11545

In [2]: print r2lang.
r2lang.cmd      r2lang.plugin

In [2]: print r2lang.plugin
```

untitled

Recorded by pancake

asciinema.org

RETWEETS 10 LIKES 13

2:34 AM - 2 Jun 2016

CODE?