

TRICK 2015

@jimmynguyc

What is **TRICK 2015**

- **T**ranscendental **R**uby **I**mbroglio **C**ontest for Ruby**K**aigi
- A Ruby programming contest started in 2013 and held in RubyKaigi 2015

What is **TRICK**

- Transcendental /,tran,sen'den(t)l/
 - of or relating to a spiritual or nonphysical realm
 - unworldly
- Imbroglia /im'brōlyō/
 - an extremely confused, complicated, or embarrassing situation

Goal of **TRICK**

- Write the most Transcendental, Imbroglia Ruby Program
- To illustrate some of the subtleties (and design issues) of Ruby.
- To show the robustness and portability of Ruby interpreters.
- To stabilize the spec of Ruby by the presence of valuable but unmaintainable code.

Related Contests

- **IOCCC**: International **O**bfuscated **C** **C**ode **C**ontest
- **UCC**: **U**nderhanded **C** **C**ontest
- **OPC**: **O**bfuscated **P**erl **C**ontest
- **IORCC**: International **O**bfuscated **R**uby **C**ode **C**ontest

IOCCC Winner 2014

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
```

```
#define A calloc(1, sizeof(m))
#define D m.
#define L ->
#define P malloc(
#define R &&
#define S ++
#define U for(
#define V if(
#define W L i
#define X L a
#define Y L b
#define Z L c
```

```
typedef struct T*_; struct T{ int t,i; char*a; _*b,c,d,e,f; } m,x,y; struct stat z; int u; void * H;
int K(int c){ return strchr(" \n\t",c); } void B(_ s,_ p) { s Y=realloc(s Y,S s W*sizeof(_)); s Y [s
W-1]=p; } _ C(char*d,int l){ _ s=A; s X=d?d:P 1); s W=l<0?d?strlen(d):0:l; return s; } _ E(_ s){ _ r
=A; int i=0; U; i<s W; ){ U; i<s W R K(s X[i]); )i S; int j=i; U; j<s W R !K(s X[j]); )j S; V j-i)B(
```

```

r,C(s X+i,j-i)); i=S j; } return r; } int F(_ a,_ b){ return a W==b W R !memcmp(a X,b X,a W); } void
l( _ s,char c ) { char*d=P 1); *d=c ; B(s,C(d,1) ); } void
J ( char*d,int l ,int o,_ v){ _ n= C(d,l); _ w; U l= 0; l
<D d W; ) { w= D d Y[l S ]; V F( w Z,n)){ V o>w W return; goto O; } }
B( D d,w= A ); w Z=n; O:w W =o ; w L d=v; } int M( _ b,int p,char*t,int
g ){ _ f= A; f L e= A; f X=t; int h=g ,i=0,j=0,n=b
W; char c=0, d,*o =b X ; U; p<n; ){ t=j?"":f X ;
int k=p; _ r=H ; U; p <n; ){ int l = 0; U; p<n R o[p]==92 ; ){ l S; p S; } c=o
[p S ]; V c^10||! l%2 ||!g){ d=c== 35 R !l R!g|| c== 10 R g^2; V d||strchr(t,c
)) { r=C(o+k,p-k- d*l/ 2-1); V d R l%2) { r X[r W-1] =c^35 ?32:
c; c =0; } break; } } c=0; } V!j)B (f L e,r?:C(o+k, n-k) ); r
=A; switch(c){ case 35:j S; case 0:break; case 10:goto O; case 36:switch(d=o[p S]){ case 36:l(f L e,
d); break; default:l(r Z=A,d); goto o; case 40:case 123:r L f=f; r L e=A; r X=d^40?"$":")$"; f=r; }
break; default:V f L f){ r Z=f L e; f=f L f; o:r L t S; B(f L e,r); } else goto O; } i=f L f R g==1;
g=i?0:h; } O: x.c=f L e; x.t=c; return p; } _ N(_ s){ _ o=C(H,0); U int i=0; i<s W; ){ _ p=s Y[i S];
V p L t){ _ n=N(p Z); p=C(H,0); U int j=0; j<D d W; ){ _ w=D d Y[j S]; V F(w Z,n)){ M(w L d,0,"$",2)
; p=N(x.c); break; } } } o X=realloc(o X,o W+=p W); memcpy(o X+o W-p W,p X,p W); } return o; } _ O(_
t){ U int i=0; i<D e W; ){ _ s=D e Y[i S]; V F(s Z,t))return s; } _ s=A; s Z=t; s L d=A; s L t--; B(D
e,s); return s; } void Q(_ s){ _ d=s Z; V!stat(memcpy(calloc(1,d W+1),d X,d W),&z))s L t=z.st_mtime;
V!s W S){ u=!s L d W R s L t<0; int i=0,j,k=s L t<0; _ c=H,t; U; !u R i<s L d W; ){ _ a=s L d Y[i S]
; U j=0; !u R j<a L d W; ){ Q(t=O(a L d Y[j S])); k|=t L t<0||t L t>s L t; } V a Z W){ c=a Z; V a L
d W)J("<", 1, 0, *a L d Y); } } U j=0; !u R c R k R j<c W; ){ J("@",1,0,d); s=N(c Y[j S]); U; s W R
strchr("@+-",*s X); )s X S; u=system(s X); } } } void G(_ b){ U; b W R K(*b X); ){ b X S; b W--; } }

```

```

int main(int i,char**a,char**e){ D d=A; D e=A; D c=A; char p[]="NBLF\1Nblfgjmf\1",*q=p; U; (*q++)--;
); J(p,4,0,C(*a S,-1)); U i=0; i<3; i+=2){ U; *a; ){ q=*a; U; *q R*q^61; )q S; V*q)J(*a,q-*a,i+1,C(q
+1,-1)); else V!i)B(D c,C(*a,-1)); a S; } a=e; } stat(p+5,&z); i=z.st_size; _ b=C(H,0),c,d; b W=read
(open(p+5,0),b X=P i,i); U; u<i; ){ V b X[u]^9){ U; u<i R K(b X[u]); )u S; u=M(b,u,"=$",0); y = x;
switch(x.t){ case 58:u=M(b,u,"$",0); D f=c=A; c L d=E(N(x.c)); c Z=A; d=E(N(y.c)); U; d W--; )B(O(*d
Y S) L d,c); break; case 61:u=M(b,u,"",0); G(c=N(x.c)); G(d=N(y.c)); U; d W--R K(d X[d W]); ); J(d X
,S d W,2,c); } } else { u=M(b,S u,"$",1); l(x.c,0); B(D f Z,x.c); } } V!D c W R D e W)B(D c,D e Y[0
Z); U u=0; !u R D c W--; )Q(O(*D c Y S)); return u; }

```

TRICK 2015

- Judges

- Yusuke Endoh (@mametter. **R**uby committer. IOCCC winner)
- Koichiro Eto (@eto. Media Artist. Chairman at NicoNicoGakkai Beta)
- Shinichiro Hamaji (@shinh. The admin of anarchy golf. IOCCC winner)
- Yutaka Hara (@yhara. The author of Japanese esolang book)
- Yukihiro Matsumoto (a.k.a. matz. @yukihiro_matz. The creator of **R**uby)
- Sun Park (a.k.a. leonid. The 1st super **R**uby golfer.)
- Hirofumi Watanabe (@eban. **R**uby committer. The 2nd super **R**uby golfer)

あなたの知らない

THE WORLD OF OBFUSCATED,

超絶技巧プログラミング

ESOTERIC, ARTISTIC PROGRAMMING

の世界

遠藤 侑介

YUSUKE ENDOH

実用性を追求するだけでは
出会えない、ちょっぴり不思議な
プログラミングの世界で
一緒に遊んでみませんか？

技術評論社

TRICK 2015

- Judging process
 - Each judge scores each entry on 10 point scale
 - Judge has no rights to score own entry
 - Judge keeps each entry anonymous during judgement
 - 12 winning entries (7 judges awards, 5 top awards)
 - Judges can win only one award

“Most Illusionistic”

```
eval(%w{require'zlib';eval(Zlib:
t+/QsfRAVrCYFw7TnrcTWauD/fW5zBch
tJJW+lba/exJ2PtUhqPhVTC8oHnwSS6
BAFLs4GnAwjdt9g7bnK2HbG/8Tsuy+s
fYz7lXyaWsFlKIm31bIxihf+xioWgvC
N5cKbJ+U6GozIsH1pGGLY3iDe4sGSPT
Fj5AgCi+HTn1Xnc6yjlSaMkQ03x4Cgz
Fv88
H1E0
dJb
S2W
tKT
EHq
bMJ
vq
KD
jk
3s
fg
4r
68bCc5rNJC7/gMCHzQZK+C2EJ52pKNEtifQ/rFhrSVmU0j/HYubQq1RZ+ztr97p9PBzwbbi1L6LXHz6
IbLNabP8D7Sdt6A=='.unpack('m')[0]))###^--- FOLLOW THE BALL ###trick2015}*")

TmXkXvlsL8QQ
v8L+YrxuF7EQ
yMtBKhrRtr5PE
48mc+HuzkUJK
6G0B3QLyHAC7
QY01CCytgT+Bf6
XntuIb15R7qtpQ
x7hfNHQLyxmnlG
8pp8nwC74Ti/K6
mVpUAbFtd
MQCAI+240
Ri4nwzeT0
FLcEQaddl
Xo0Z8+H4hnGR
CtH8R8sd5jcSw0
exUiRY4efSdnUC
5vAoLrMfYUiaxD
R7BJCBncL4C5Ii
/85oJEyJUclfaa
T4KjGp9N1a8rbXv
:Inflate.inflate('eJzNVt9zm0gMf
9jrQGIDweBQ//jft9Kubey0TqYvvX1Y
hT0BP80EvqJJ8YvcduIZSCPLhCISA
c2wcueAhsQN9Fz66MHBh6MJlzMZdV+
Tuk1o/Y88qFKxb6WqBoykaAdlsGXp3
AD0wLYB04DBYcENG4iP2bDpnXh2wai
oMD9Mocp+FuQbggxfAKWrY+Z4PQxDi
DleXXxd4NJ
JHt7Ps7vr6
89ryv51GSVtF
LS4fqU2GqBkt
b3dRrjvgftU8
1edhdEaW++hy
Xo0Z8+H4hnGR
CtH8R8sd5jcSw0
exUiRY4efSdnUC
5vAoLrMfYUiaxD
R7BJCBncL4C5Ii
/85oJEyJUclfaa
T4KjGp9N1a8rbXv
CDm
bzm
YY
N8
GH
HL
g0
U
3
A
G
f
A
```

by Don Yang (USA)

Demo

Key **TRICKs** / Exploits

- zlib-compressed
- Base64 Encoded
- Can't really see anything :(

“Most **I**nconsistent”

```
p (?\\A-p +  
  ?\\B-p +  
  ?\\C-p +  
  ?\\D-p +  
  ?\\E-p +  
  ?\\F-p +  
  ?\\G-p +  
  ?\\H-p +  
  ?\\I-p +  
  ?\\J-p +  
  ?\\K-p +  
  ?\\L-p +  
  ?\\M-p +  
  ?\\N-p +  
  ?\\O-p +  
  ?\\P-p +  
  ?\\Q-p +  
  ?\\R-p +  
  ?\\S-p +  
  ?\\T-p +  
  ?\\U-p +  
  ?\\V-p +  
  ?\\W-p +  
  ?\\X-p +  
  ?\\Y-p +  
  ?\\Z-p).b  
  .tr 'A-Z', ('_')/~0
```

```
BEGIN{class Object ; def  
  -(o)self end;alias/ -end}
```

by Koichi Sasada (Japan)

Demo

Key **TRICKs** / Exploits

- Character literal **#?**
- Escape Sequences **\C-X**
- Convert to ASCII-8BIT **String#b**
- Range 'A-Z' with **String#tr**
- Reserve word **BEGIN**
- Best way to write underscores **('_')/~0**

“Best D Document”

```
Description from Ky.
-----
What if $1 and under junk games take over the world. It would makes you crazy.
We must fight against the future and will soon launch a new fantastic action game named TRICK!!!

Demo
-----

    2020-03-25
    Criminal 1 :: There he is! Kill the traitor.
    Agent 1    :: Cover the man until 006 and 007 rescue us.
    Agent 2    :: Okay. Come here!
    Criminal 2 :: You must die if you bother us.

How to Get Demo Started if not for path in ENV
-

1.1
next __LINE__

    ruby -v
The command shows ruby version like this:

    'ruby 2.2.3p173 (2015-08-18 revision 51636) [x86_64-darwin14]'

1.2
next __LINE__

    ruby entry.rb

end
```

by Yoshi Taka (Japan)

Demo

whitequark/parser



AST



mbj/unparser

Description from Ky.

What if \$1 and under junk games take over the world. It would makes you crazy.

We must fight against the future and will soon launch a new fantastic action game named TRICK!!!

Demo

2020-03-25

Criminal 1 :: There he is! Kill the traitor.

Agent 1 :: Cover the man until 006 and 007 rescue us.

Agent 2 :: Okay. Come here!

Criminal 2 :: You must die if you bother us.

How to Get Demo Started if not for path in ENV

-

1.1
next __LINE__

ruby -v
The command shows ruby version like this:

'ruby 2.2.3p173 (2015-08-18 revision 51636) [x86_64-darwin14]'

1.2
next __LINE__

ruby entry.rb

end

```
if (($1 &&
under(junk(games(take(over(the(world.It(would(makes(you(crazy.W
e(must(fight(against(the(future)))))))))))) &&
will(soonestart(a(new(fantastic(action(game(named(TRICK!(!(!
Demo))))))))))
```

```
Description(from(Ky.- - (-(-(-(-(-(-(-(-What))))))))))
```

```
end
```

```
(((-(-(-(-(-(-(-(-(-2020)))))))))) - 3) - 25
```

```
until (6 && 7)
```

```
Criminal(1.There(he(is!
(Kill(the(traitor.Agent(1.Cover(the(man))))))))))
end rescue us.Agent(2::Okay.Come(here!))
```

```
if (!for path in ENV do
-1.1
next 19
ruby(-v)
The(command(shows(ruby(version(like(this: "ruby 2.2.3p173
(2015-08-18 revision 51636) [x86_64-darwin14]"))))))
1.2
next 27
ruby(entry.rb)
end)
```

```
if you(bother(us.How(to(Get(Demo(Started))))))
Criminal(2.You(must(die)))
end
```

```
end
```

Key **TRICKs** / Exploits

- Continuing statement without terminating (. :: ! -)
- until ... rescue ...
- `__LINE__`

“Most **T**imely”

```
0.times{%w{
  2422-02-10T21:45:38+09:00
  2580-06-19T08:53:09+09:00
  2233-01-20T02:06:42+09:00
  2535-06-28T10:04:32+09:00
  2365-05-11T19:00:00+09:00
  ...
  2341-10-04T21:21:57+09:00
  2751-11-28T05:17:41+09:00
  2463-06-05T08:36:01+09:00
  2375-04-30T18:11:29+09:00
  2069-06-26T10:44:16+09:00
  2876-12-06T18:46:16+09:00

  "Back to the future?"

}}

GC.start

require'time';s = []; ObjectSpace.each_object{|u|s<<u}
eval s.map{|a|(((t=Time.iso8601(a))>Time.now)? t.to_i:
$')rescue nil}.compact.sort.uniq.map{|t|(t&0xff).chr}.
join# Quoted from "Back to the Future Part III (1990)"
```

by Koichi Sasada (Japan)

Demo



Key **TRICKs** / Exploits

- ObjectSpace (requires GC.start)
- Bitwise operator **&**
- Ruby does not suffer from year 2038 problem
(Since Ruby 1.9.2, Time implementation uses a signed 63 bit integer)

“Matz Lisp”

```
'(#|'.b;module Scheme;T,R,I,C,K=Struct.new(:a,:d,:o){include Enumerable#|#)#|
def initialize x,y=(),o=0.!.super c(x),c(y),o;K.empty?&&(R.delete a;R.delete d
R<<self)end;def-@;T[:-,self]end;alias == equal?;def call i;Scheme.t self,i end
def to_a;[T[:*,self]]end;def c o;Array===o ?Scheme.l(o):o end;def each&b;b.(a
d&d.each(&b)end},[],e=Struct.new(:t,:u){def[]=k,v;t[k]=v end;def[]i;t.fetch(i
){u[i]}end},e[_h={}],[];def Object.const_missing i;i end;class::Fixnum;undef[]
def[]i;T[self,T[i],1]end end;module A;def call *a;a.empty?? T[self]:
Scheme.t(self,a[0])end;def-@;T[:-,T[self]] end end;class::Array;def
call a;T[self,T===a&a.o ? a:T[a],1]end end;class::Symbol;def*i;T[
self,T[:*,T[i]]]end;include A
T[self,T[:-,T[i]]]end end;at_exit{
|l|v l}};B=TOPLEVEL_BINDING;class::
B);def m.method_missing i,*s;a,s;
end;class<<self;def e f,n;f.map{
e x.d.d,I[f ?Hash[f.zip a]:{},{},n]}
){|a,i|T[i,a]}end;def t s,i;T[s,
i:T[i],1]end;def v x,n=C;case
r x,x.d.a,n when :LET;e={};x.
e x.d.d,I[e,n]when :IF;v(v(x
: ( ),n)when :DEFINE;Symbol.
n.[]=u.a,r(x,u.d,n))when
true)?TRUE:d when :COND
n if :ELSE== x.a.a||v(x
v i,n); f.call(*r);end
end;%w(CONS a,b T[a,b]
l l.to_a.size APPLY f,*n,s f.(*(n+(s||[]).map.to_a)) SYMBOL? o Symbol===o READ
* t=""*1;begin;gets&&eval(t<<($_!=$/?$_:x),B);rescue(Object);retry;end MAP p,l
l(l.map(&p)) NUMBER? o Numeric===o STRING? o String===o NOT o 0.!.==o ERROR *s
fail(s*"s") EOF-OBJECT? o o==() DISPLAY o puts(o) SET-CDR! p,o p.d= SET-CAR!
p,o p.a= NULL? o o==() EQ? a,b a.equal?b * *s s.inject:* MIN *s s.min LIST *s
l(s) CAR p p.a CDR p p.d).each_slice(3){|t|eval'_h[:"%s"]=->%s{%s}'%t}end;#|#

(DEFINE (FACT N) . (
  (IF (EQ? N . (1)) . (
    1 [* N . ((FACT (- N . (1))))]))))

(DISPLAY (FACT 6))
```

by Kazuki Tsujimoto (Japan)

Demo

Key TRICKs / Exploits

- Polyglot code (Ruby + Scheme)

As Ruby

```
def initialize x,y,z;super r(x),r(y),o;k.empty?MA(R.delete a;R.delete d  
xxxxx)end;def-@;T[-,self]end;alias == equal?;def call l;Scheme.t self,l end  
def to_a;T[*self]end;def c o;Array==o ?Scheme.l(o)o end;def each&b;b(o)  
MA2.each&b)end;[[e-Struct.new(t,u){def[]-k,v;t[k]-v end;def[]t.fetch  
[u[i]]end;e[-_]=[];}};def Object.const_missing l;l end;refine(Planus){def[]l  
[self,T[l],l]end;module M; def-@;T[-,self]end;def call *a;A  
empty? T[self];Scheme.t(self ,a[0])end;end;refine(Array){def call a  
[self,T==a&a.o ? a-T[a],l  
[self,T[*],T[l]]end; def-l;  
T[-,T[l]]end;};at_exit{R.drop(k[  
refine(String){include A; refine  
s.s.empty? l;T[T,T==a&a.o ? a;T  
Array==l ?T[l[0]]:l;end;M.o ? l;  
[l]-l;end;};at_exit{R.drop(k[  
def l v,v.reverse.inject([]){[a,  
when l;case x.a when :LORDA;r  
x.a.each{|l|e[l,a]-v l,d;end;end;  
x,n)N.d.d.at (y=x.d.d.d)ty. a t  
break o x.a.d,n if:ELSE=va. a a []  
symbol==(u,x,d,a) ?(N[u]=  
QUOTE;d=x,d,a,d=true ?  
f,*x.map{|l|v l,n};f.  
x;end;end;end;Be(PAIR?  
x.dno LIST ?s l(s) CAR  
o,p,aoo ERROR *s fail(s""Vs") READ * to"using(Scheme);"  
to(S,l-$/TS,ia),TOPLEVEL_BINDING);rescue(Object);retry;end NULL? o o-() CONE  
a,b T[a,b] APPLY f,*n,s f.('ne(s)[]].map.to.a)) EOF-OBJECT? o o-() LENGTH l  
l.to.a.size SYMBOL? o Symbol==o EQ? a,b a.equal?b DISPLAY o puts(o) CDR p,p,d  
* *s.s.inject+ - *s s.inject+ MAP p,l l(l.map&p)) STRING? o a.l.s.at(0)String  
ROR *s s.dn).each slice(0,10).eval hl:"R["<=KaKa]N)end;using Scheme.l
```

DSL interpreter

```
(DEFINE (FACT N) (IF (EQ? N 1) 1 (* N (FACT (- N 1)))))  
(DISPLAY (FACT 6))
```

Scheme-like Ruby DSL

As Scheme

```
def initialize x,y,z;super r(x),r(y),o;k.empty?MA(R.delete a;R.delete d  
xxxxx)end;def-@;T[-,self]end;alias == equal?;def call l;Scheme.t self,l end  
def to_a;T[*self]end;def c o;Array==o ?Scheme.l(o)o end;def each&b;b(o)  
MA2.each&b)end;[[e-Struct.new(t,u){def[]-k,v;t[k]-v end;def[]t.fetch  
[u[i]]end;e[-_]=[];}};def Object.const_missing l;l end;refine(Planus){def[]l  
[self,T[l],l]end;module M; def-@;T[-,self]end;def call *a;A  
empty? T[self];Scheme.t(self ,a[0])end;end;refine(Array){def call a  
[self,T==a&a.o ? a-T[a],l  
[self,T[*],T[l]]end; def-l;  
T[-,T[l]]end;};at_exit{R.drop(k[  
refine(String){include A; refine  
s.s.empty? l;T[T,T==a&a.o ? a;T  
Array==l ?T[l[0]]:l;end;M.o ? l;  
[l]-l;end;};at_exit{R.drop(k[  
def l v,v.reverse.inject([]){[a,  
when l;case x.a when :LORDA;r  
x.a.each{|l|e[l,a]-v l,d;end;end;  
x,n)N.d.d.at (y=x.d.d.d)ty. a t  
break o x.a.d,n if:ELSE=va. a a []  
symbol==(u,x,d,a) ?(N[u]=  
QUOTE;d=x,d,a,d=true ?  
f,*x.map{|l|v l,n};f.  
x;end;end;end;Be(PAIR?  
x.dno LIST ?s l(s) CAR  
o,p,aoo ERROR *s fail(s""Vs") READ * to"using(Scheme);"  
to(S,l-$/TS,ia),TOPLEVEL_BINDING);rescue(Object);retry;end NULL? o o-() CONE  
a,b T[a,b] APPLY f,*n,s f.('ne(s)[]].map.to.a)) EOF-OBJECT? o o-() LENGTH l  
l.to.a.size SYMBOL? o Symbol==o EQ? a,b a.equal?b DISPLAY o puts(o) CDR p,p,d  
* *s.s.inject+ - *s s.inject+ MAP p,l l(l.map&p)) STRING? o a.l.s.at(0)String  
ROR *s s.dn).each slice(0,10).eval hl:"R["<=KaKa]N)end;using Scheme.l
```

A comment ignored

```
(DEFINE (FACT N) (IF (EQ? N 1) 1 (* N (FACT (- N 1)))))  
(DISPLAY (FACT 6))
```

Normal Scheme code

“Most **F**ragile”

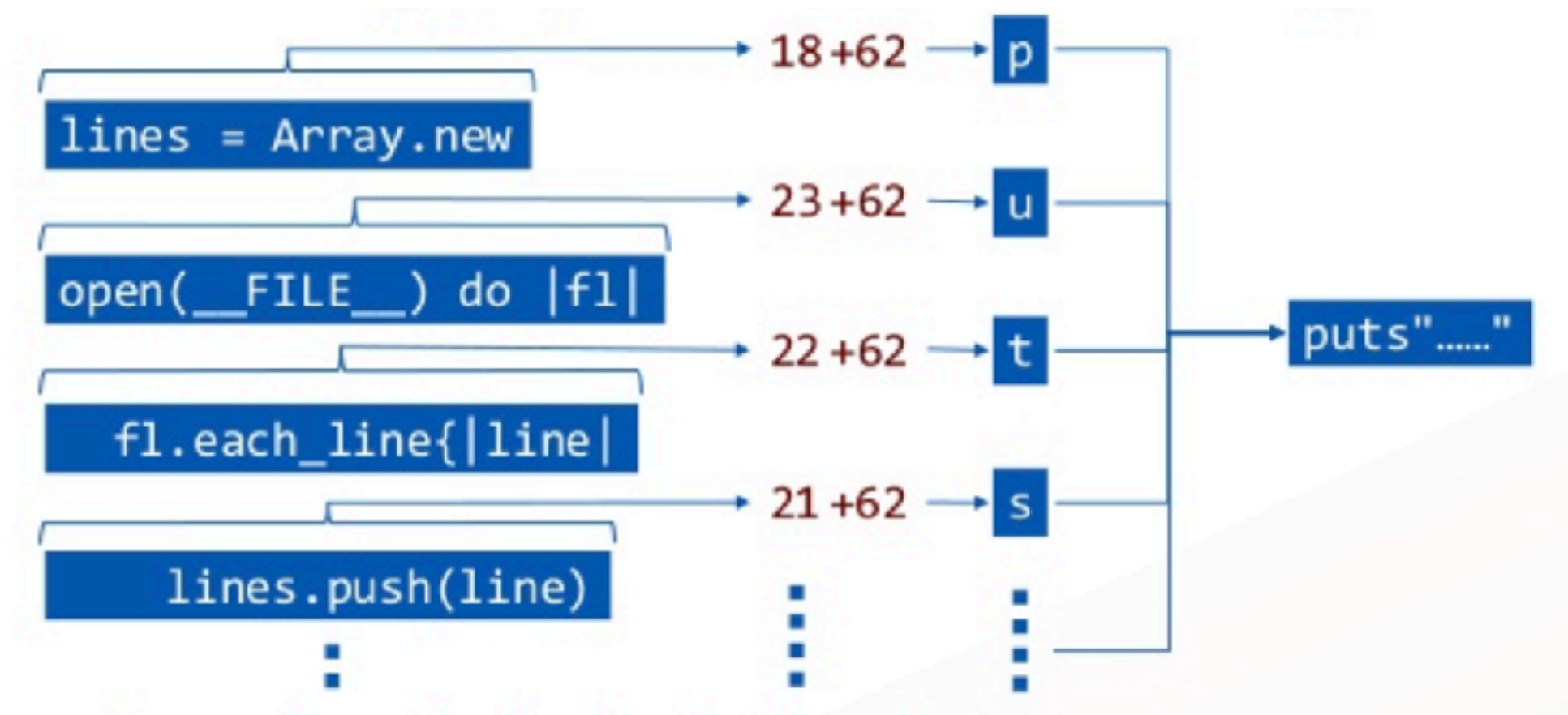
```
lines = Array.new
open(__FILE__) do |fl|
  fl.each_line{|line|
    lines.push(line)
  }
  m=15+15
  n=62
  $e = lines.
    map do|ln|
      sz = ln.size
    ;
      (sz<5?sz+m:sz+n).
      chr().swapcase
    end.join
  $e
  eval "#$e"
end
```

by Nakamura Usaku (Japan)

Demo

Key TRICKs / Exploits

- Code with line length



“Most Beautiful Pattern”

```
-> ( &_ ) { [ - , ?e +  
    ?v + ?a + ?l , q = ?e +  
    ?v + ?a + ?l + ?( + ?d +  
    ?[ + ?1 + ?9 + ? , + ?9 +  
    ?9 + ?9 + ?] + ?* + ?" +  
    ?" + ?) ] &(s , *d = %I{  
#{ ?s + ?e + ?n + ?d } -> ( &_ ) { [ - , ?e + ?v  
    + ?a + ?l , q = a= d[ 0,  
    19 ]; ;b =q .c ha rs .m ap { |  
    _| b& &a << ?+ ;a << ?? +_ ;b  
    |= 1} ;a += %q !] X} [X &( sX  
    ,X *d X= X% I{ X# {X ?s X+ X? e  
    X +X ?n X+ X? dX }X !. sp li  
    t( ?X )+ d+ " } X) X[ X0 X] X]  
    ". sp li t( ?X ); pu ts (a .m  
    ap (& :t o_ s) .p ac k( "A 6"  
    *a .s iz e) .u np ac k" A6 1"  
*1 9) #T RI CK } ) [ 0 ] ]
```

by Shinichiro Hamaji (Japan)

Demo

Key **TRICKS** / Exploits

- Quine
- Parser ignores spaces
- Problem: No room for “eval”
- Solution: Exploit symbol array literal

```
p %I{ foo bar } #=> [:foo, :bar]
```

- Detailed solution in Mame's book - Chapter 6

“Most General S Solver”

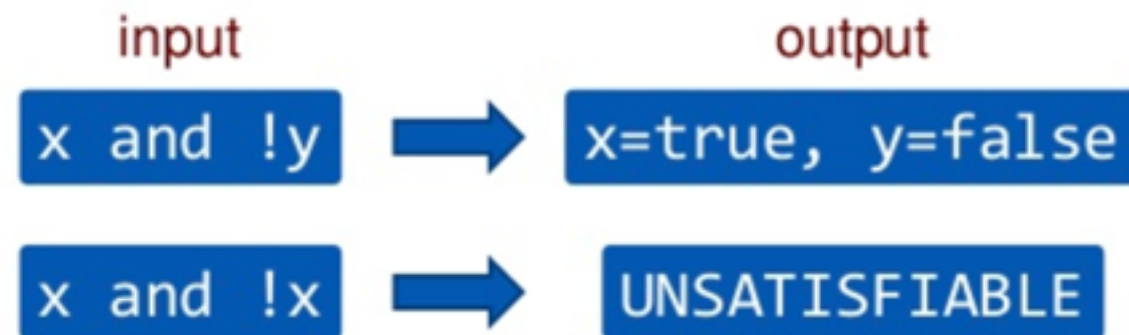
```
_='s %sSATISFIABLE';puts eval$<.read.gsub(/.*p.*?(\d+).*?$/\d+/m)
{$1?%w[?-* +'=-'~/#{'(-?)* }-*(?=)*$1:$&>?0?"\#$&$|":'$)(?='}
+' )/x?[_%p%i=0,[*$~].map{|x|x>?-?:v:eval(x+?1)*i-=1}*"" "]:_%:UN'
```

by Keisuke Nakano (Japan)

Demo

Key **TRICKS** / Exploits

- SAT Solver



- Solve by translating into Regex matches

“Least General S Solver”

```
class String;def[]*a;$*<<a;b;end;end;
_=0;z="C=Fiber;s=$*;a=*0..8;l=C.new{e
xit},*a.product(a).select{|r,c|s[r][c
]==0}." [1,9,_,_,_,8,_,_,5]+"map{|r,
c|C.ne" [_,_,2,_,5,_,_,8,9]+"w{o=s[r
][c];l" [8,_,6,7,4,_,_,_,_]+"oop{(1.
.9).map{|n|C.yield(s[r][c]=n)if a.non
e?{|k|" [_,_,_,_,_,4,_,9,2]+"s[r][k]
==n||s" [_,2,3,_,7,_,8,1,_]+"[k][c]=
=n||s[" [5,6,_,8,_,_,_,_,_]+"r-r%3+k
%3][c-c%3+k/3]==n}};s[r][c]=o;C.yield
}}},C." [_,_,_,_,_,2,7,9,_,3]+"new{loo
p{puts" [9,3,_,_,8,_,1,_,_]+" s.map{
|r|r*" [2,_,_,5,_,_,_,4,8]+" '}<<'
;C.yield}};c=l[i=1];loop{c=l[i+=c.res
ume ? 1:-1]}}";eval z.tr ?\n, ''
```

by Benoit Daloze (Belgium)

Demo

Key **TRICKS** / Exploits

- Sudoku Solver
- Mixing data with code

```
class String;def[]*a;$*<<a;b;end
```

```
code="pri"[1, 2, 3]+"nt 1"
```

```
code="print 1"
```

```
$*=[[1, 2, 3]]
```

“Doubling **A**mphisbaena”

```
;; ;;      ;; ;;
;; ;;      ;; ;;
;;eval$s    =%q[i=1#
eval(%q[     xxxxxxxx
xx  xxxx  xx  xx  xxxx  xx
xx  xxxx  xx  xx  xxxx  xx
xxxxxxx     xxxxxxxx
xxxxxxx     xxxxxxxx
xx  xx  xxxxxxxxxx  xx  xxxxxxxx
j, t, p=0,[?;], " ev al$s=%qx
[$s]".split*"";i,j,t=i-j,i+j,(x
[b=?\s]*j.abs+t).map{|s|r=t.shix
ft ||b;r.gsub!(?;){p.slice!0}if $x
f| |=p>p=p.center(i*i+j*j,?);r ,x
s=[s,r]if(i*j<0);(b*i.abs+s).ljsx
ust(r.size).gsub(b){r[$`.size]|x
|b}}unti l$ f;puts(t)# xx xx
xxxxxxx  xx  xxxxxxxxxx  xx  xx
xxxxxxx  xxxxxxxx
xxxxxxx  xxxxxxxx
xx  xxxx  xx  xx  xxxx  xx
xx  xxxx  xx  xx  xxxx  xx
xxxxxxx  x].gsub\
/x.*|\s/ ,"" )#];;
;; ;;      ;; ;;
;; ;;      ;; ;;
```

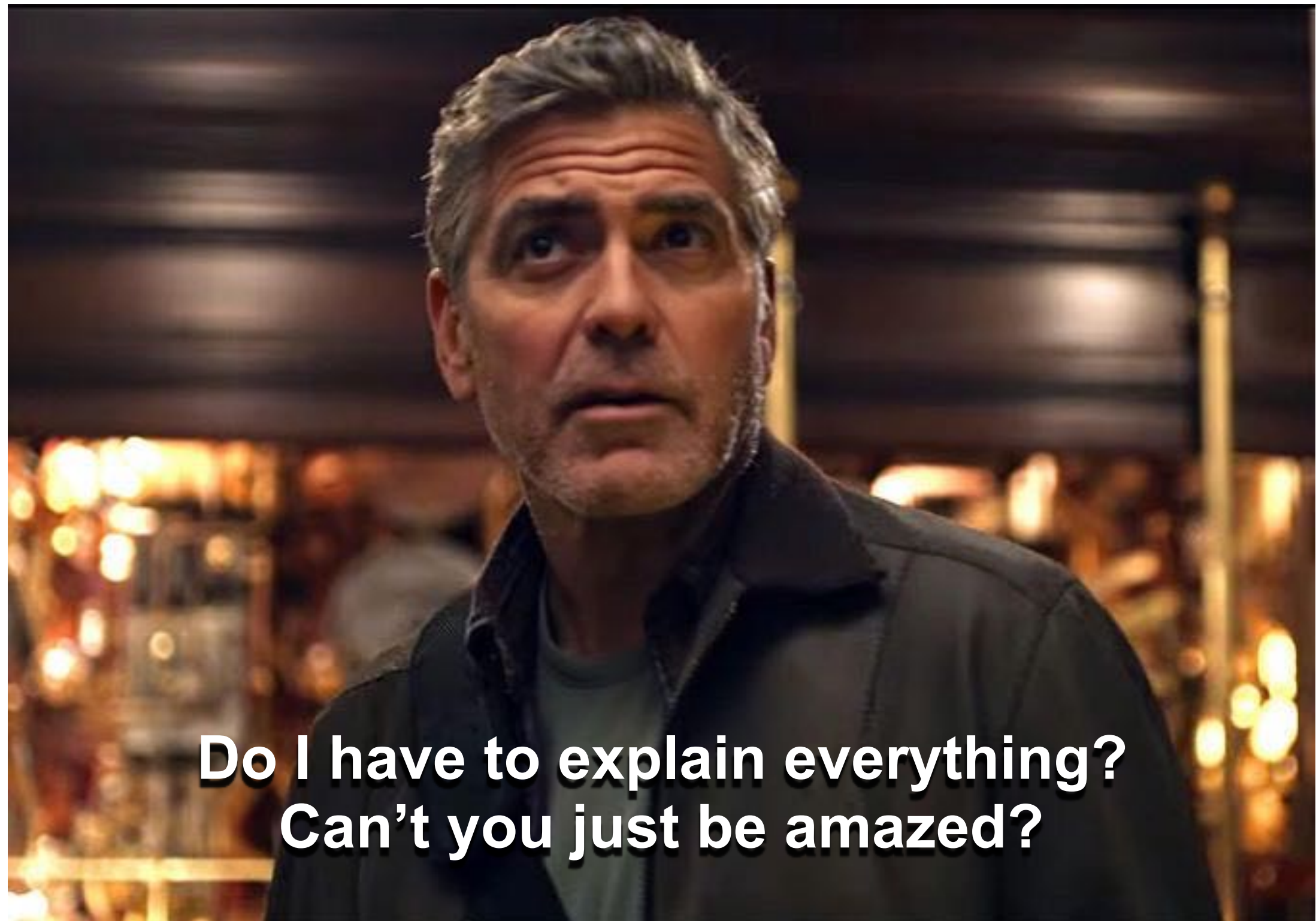
by Monae (Japan)

BRONZE

Demo

Key **TRICKs** / Exploits

- Amphisbaena - mythical twin head serpent
- Quine



**Do I have to explain everything?
Can't you just be amazed?**

“Most **R**eadable ALU”

```
%%while eval '_=%%r%%(.)... \1=%%
%=~[%%%,%%%,%%s  ?=]*%%%%
%#"']*%%%%3x+1?%%' .%  %%",%*p(_||
=eval($**%%))
```

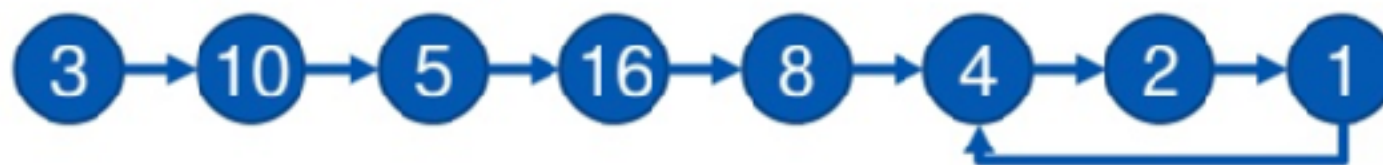
by Keisuke Nakano (Japan)



Demo

Key **TRICKs** / Exploits

- Collatz Sequence
 - if even, $n = n / 2$
 - if odd, $n = 3n + 1$



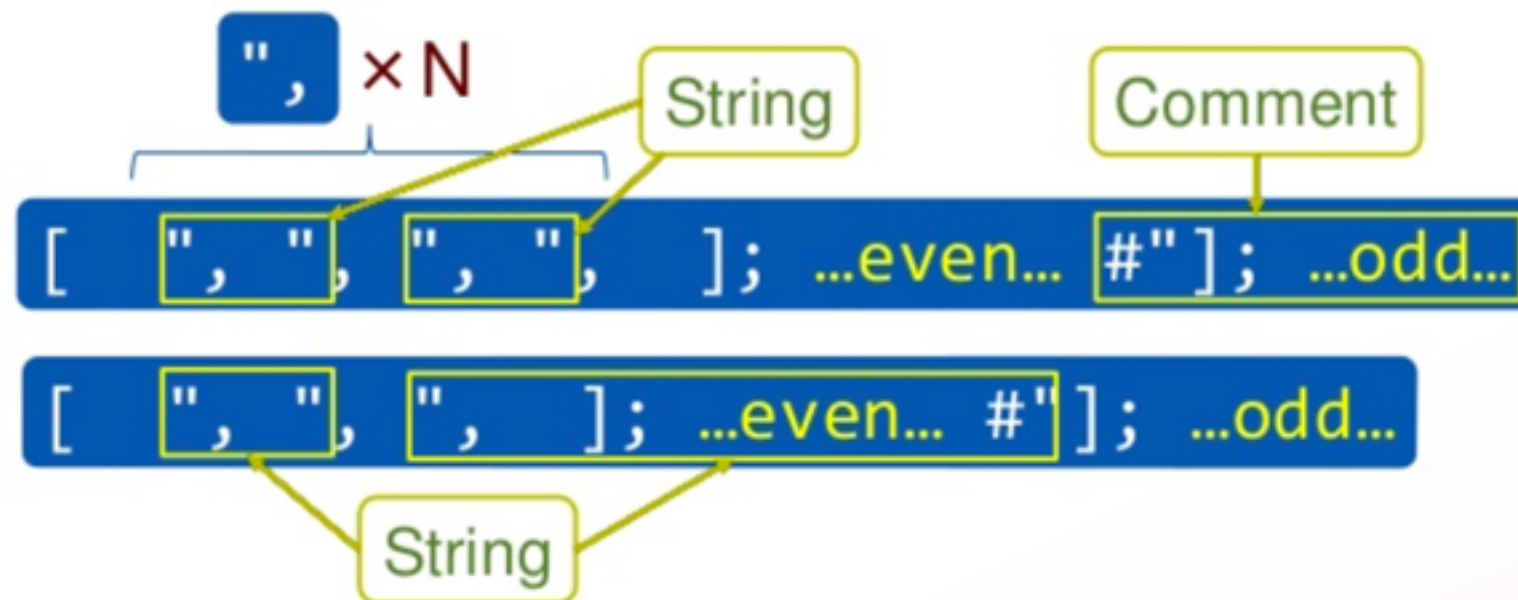
Collatz conjecture: Any n will eventually boil down to 1.

- Uses no branches (if .. else)
- Uses no integer arithmetics



Key **TRICKS** / Exploits

- Branching without if ... else



Key **TRICKS** / Exploits

- Integer arithmetics with Regex match operator `=~`

```
n = (/(.+)\.\.\.\1=/ =~ eval('[",,,,,"'+ '",'*n + '   ?=].join#"].join("3x+1?")'))
```


“Best Piphilology”

```
big, temp = Array 100000000**0x04e2
srand big
alias $curTerm $initTerm

Numeric
Interrupt

big += big
printout _pi_ finish if $never
init ||= big
$counter ||= 02

...
```

by Kinaba (Japan)



Demo

Key **TRICKS** / Exploits

- Piphilology - the creation and use of mnemonic techniques to remember a span of digits of π

Yes I have a number
3 . 1 4 1 6 $\approx \pi$

big , temp = Array 100000000 ** 0x04e2
3 . 1 4 1 5 9 2 6

- Calculates Pi up to 10000 digits

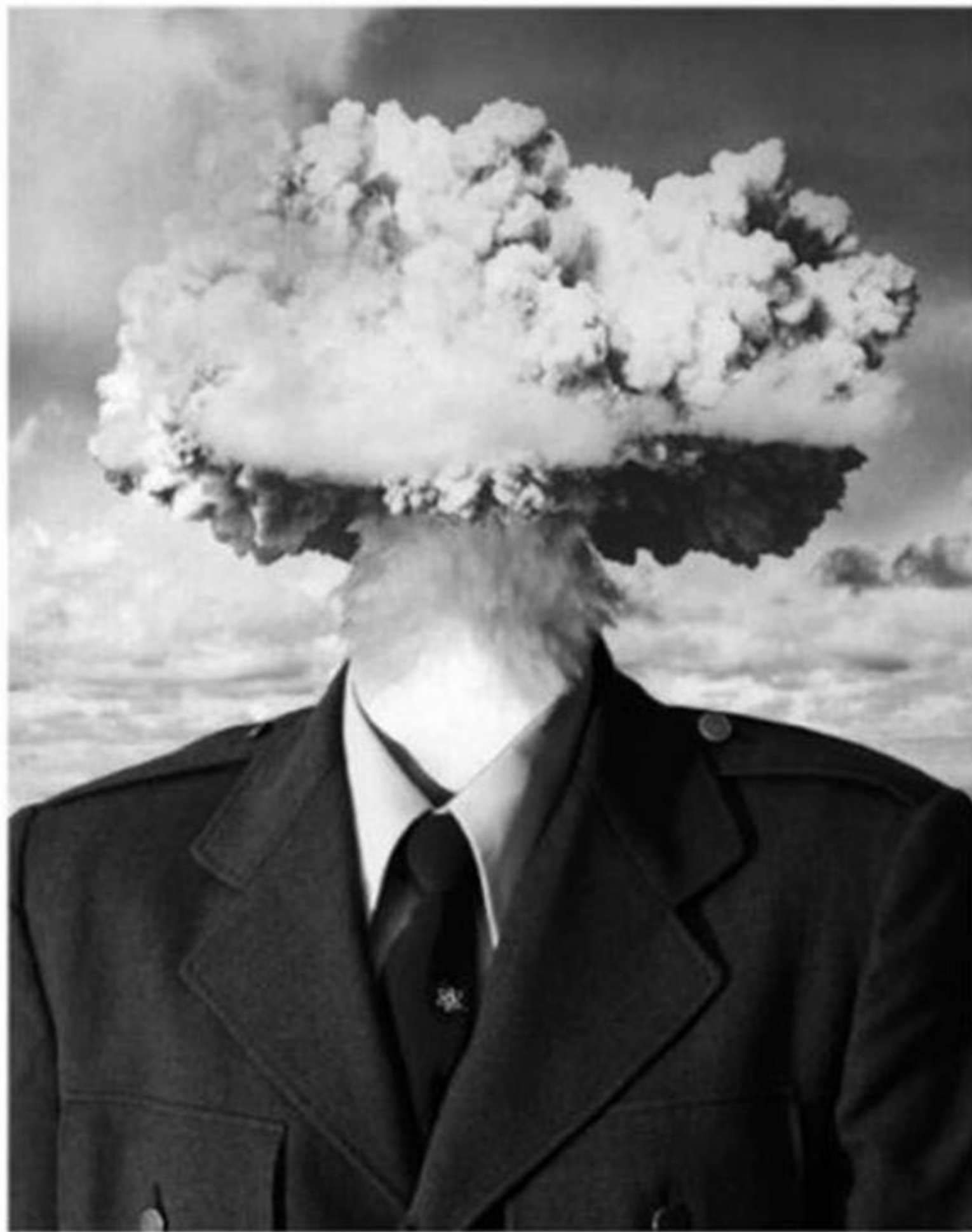
Key **TRICKs** / Exploits

- Abuse “**alias**” of global variables \$curTerm and \$initTerm to access same value
- Use **srand** as storage

Key **TRICKs** / Exploits

- 77 essential tokens
- 165 no-op fillers
- Blow up ratio

$$\frac{165+77}{77} \approx 3.14$$



Prizes

- Title of “Insane” among Ruby community
- Top 5 entries will be checked into Ruby’s official repository as “samples”

Final T Thoughts

- Programming should be fun
- Even Matz's constantly amazed by what people do with Ruby
- Enjoy programming != finding out which language is “fastest”
- So stop asking “is Ruby slower than XXX?”
- Coz frankly we don't give a shit 🙄

Google Search Term:
“trick 2015 rubykaigi”

<https://github.com/tric/trick2015>

Thank You