

Linux Day

2010

L'Anello di Congiunzione tra L'Informatica e le "Bionde": il Perl Offuscato!

*"Pole la donna permettisi di... comprendere il codice offuscato? **Si!**"*

*Filosofia, tecniche ed esempi pratici per una quasi completa
(in)comprensione di due mondi apparentemente dissimili ma dai
connotati stupefacentemente comuni."*



Per poter seguire il seminario:

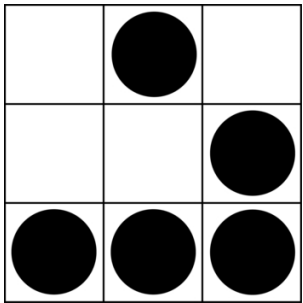
1. **NON** è necessario conoscere il Perl.
2. **NON** è necessario aver conosciuto una "bionda".

Who's who

Gianfranco Ciotti – *member of AGOW and orgoglioso to be!*

```
#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>);my
my$camel :while(
  ->Dromedary
  t(@dromeda
  SCAMEL...;if(d
  $camelhump++1
  @camelhump);&&/\S/){$camelhump+1<<$CAMEL;}
defined($_=shift(@camelhump)&&/\S/){$CAMEL=$camelhump+1<<$CAMEL;
L..)$camel="n";}@camelhump=split(/\n/,$$);foreach($
ump);$camel($chomp;$CAMEL=$y/LJF7/);foreach(@
camelhump){chomp;$CAMEL=$y/12345678/J
064\065\066\067\070/;foreach(@
print"$_040$CAMEL\n";}
/LJF7\173\175\047\12345678/\n
...print"040$_$CAMEL\n";undef$_;
...;
...al("seek\0400
0:");foreach(1..3)
@camelhump;my$camel:
<DATA>){$_=sprintf("%-6
1=split(/");if(defined($
p=split(/");while(@dromeda
my$CAMEL=3;if(defined($_=shif
)$camelhump+1<<$CAMEL;)}
$camelhump+1<<$CAMEL;)}
if(defined($_=shift(@
dromedary1))&&/\S/){
CAMEL...;if(d
$camelhump+1<<$CAMEL;)}
if(defined($_=shift(
@camelhump)&&/\S/){$camelhump+1<<$CAMEL;}
defined($_=shift(@camelhump)&&/\S/){$CAMEL=$camelhump+1<<$CAMEL;
L..)$camel="n";}@camelhump=split(/\n/,$$);foreach($
ump);$camel($chomp;$CAMEL=$y/LJF7/);foreach(@
camelhump){chomp;$CAMEL=$y/12345678/J
064\065\066\067\070/;foreach(@
print"$_040$CAMEL\n";}
/LJF7\173\175\047\12345678/\n
...print"040$_$CAMEL\n";undef$_;
...;
...$_=reverse;
CAMEL=$_;y
73\0 47/\n
eva
)::;s
\124
1 41\0
141
6 3\
156
120\1
040\14
\145\
0\157
122\1
\040
3\16
3\056
4\040\
40\160\
\155\151
\151\1
```



“But did you [...] ever take a look behind the eyes of the hacker? Did you ever wonder what made him tick, what forces shaped him, what may have molded him?”

I am a hacker, enter my world...”



The Mentor, 1986

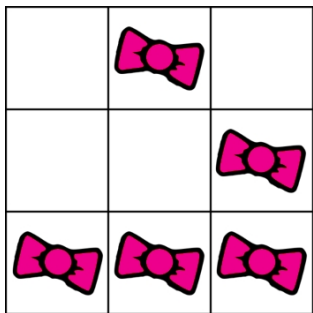
Linux Day

2010

Who's who

~~Gianfranco Ciotti~~

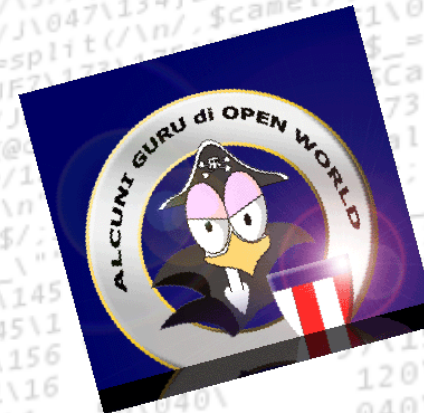
Jane "Perla" Blonde – member of AGOW too...
and orgogliosa to be, (re-too)!



"Oops! ...I Did It Again

yeah yeah yeah yeah yeah
yeah yeah yeah yeah yeah yeah "

Britney Spears, 2000



```

#!/usr/bin/perl -w
use strict;

ATA,0.
(<DATA>:)my
my$camel :while(
9s" $_) .my@dromedary
-><DATA>)) (@camelhump
cy1)(my$camelhump=0
@<dromedary1
...:if(d
$camelhump)+1 <<$camelhump)
&&/\S/){$camelhump+1<<$CAMEL;}
L:)}$camel ="\n";$y=LJF7
ump);}$camel={chomp;$camel=$y/LJF7/
camelhump}(chomp;$camel=$y/12345678/
064\065\066\067\070/:y/12345678/@
print"$_1040$camel\n";}foreach(@
/LJF7\173\175\1047/12345678/:y/1
$_=reverse;print"1040$_$camel\n
("seek\040DATA,0,0;");undef$/;
^X...:map{eval"print"$_1
50\145\040\165\163\145
40\143\141 155\145\1
\147\145\0 40\151\156
257\143\ 151\141\16
1040\167 154\040\15 1\163\
45\162\ 64\162\1 41\144
1\162\ 153\04
040\11 7\047\
1\154\1 54\171
45\15 012\101\16 1\14
1046\ 7\143\15 1054
3\15 4\145\163
1\16 1\111\156\14 3\056
1040 125\163\145\14 4\040\
1040\ 51\164\1 50\0 40\160\
167\1 \155\151
145\162 \151\1
\163\163
57\156\056

```

Linux Day

2010

Who's who



Jane Blonde

```
perl -e '+$,=+$$";+s//"\160\162\151\156\164\x20\x27\
\x4a\x61\x6e\x65\x0a\x27"/ee'
```


Linux Day

2010

`$_ = "Doverose Precisazioni";
s/ve/lo/; print`



NON

parleremo di...

Linux Day

010

\$_ = "Doverose Precisazioni";
s/ve/lo/; print

camel code



... questo
"anello di congiunzione"

```
#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>);my
my$camel;while(
=>DATA>){my$camelhump;
t(@dromedary1
SCAMEL--);if(d
$camelhump+1
$camelhump)&&/\S/){$camelhump+1<<$CAMEL;
de://3camelhump
ump);$camel{chomp;$camelhump;my$CAMEL=3;if(defined($
064\065\066\067\070//y/12345678/JL7F\175\047\173\175\047\134)L^7FX")){
print"$_\040$CAMEL\n";}foreach(@camelhump){$camelhump+1<<$CAMEL;
/LJF7\173\175\047/12345678//y/12345678/JL7F\175\047\134)L^7FX")){
$_=reverse;print"\040$_$CAMEL\n";}s/\s*///g;eval($_);s
("seek\040DATA.0.0;");undef$/;$/={}/g;DATA__
^x...;map[eval"print"$_\040\163\145\040\157\1
11 50\145\040\165\163\145\040\157\1 46\040\1 41\0
40\143\141 155\145\1 54\040\1 51\155\ 141
\147\145\0 40\151\156 \040\141 57\156
257\143\ 151\141\16 4\151\1 120\1
\040\167 \151\164\1 50\040\ 040\14
45\162\ 64\162\1 41\163\ \145\
1\040\1 1\162\ 153\04 0\157
155\14 1\162\ 7\047\ \040
\146\ 040\11 54\171 \040
45\15 1\154\1 54\171 3\16
\046\ 012\101\16 1\14
3\15 7\143\15 \054
1\16 4\145\163 3\056
\040 \111\156\14 4\040\
1040 125\163\145\14 40\160\
1040\ 51\164\1 50\0 \155\151
167\1 \151\1
145\162
\163\163
57\156\056
```


Linux Day

2010

Parlano di L*i...



L'anello di congiunzione: il Perl Offuscato

Linux Day

2010

Parlano di L*i...

indiscusso protagonista

The Ring Tron



Linux Day

2010

Parlano di L*i..
preziosissimo collaboratore



Professional Mutt Works Fine!

Dietrich Ayala, Christopher Browne, Vivek Chopra,
Dr. Poornachandra Sarang, Kapil Apshankar, Tim McAllister



Wrox technical support at: support@wrox.com
Updates and source code at: www.wrox.com
Peer discussion at: p2p.wrox.com

camel code

```
ev
\0400
(1..3)
$camel:
ntf("%-6
efined($
(@dromeda
ed($_=shif
=1<<$CAMEL;)
ty1))&&/\5/){
ined($_=shif
)}$CAMEL--;if(
ellhump+=1<<$CAME
L^7FX")[$camellh
:$camel):foreach(@
\047\061\062\063\
73\047\061\062\063\
p){chomp;$Camel=$_;y
JL7F\175\173\0 47~/;
\s*//g;eval: eval
A>;s/\s*//g:( );;s
)/g: DATA_ 124
67\1 46\040\1 41\0
0\1 51\155\ 141
141 \163\16 3\
1\1 57\156
40\ 120\1
63\ 040\14
44 \145\
4 0\157
122\1
\040
3\16
1\14
\054
3\056
4\040\
40\160\
\155\151
\151\1
```

Linux Day

2010

`$_ = "Doverose Precisazioni";
s/ve/lo/; print`



NON

parleremo di...

Linux Day

2010

Paura eh...



Linux Day

2010

queste “bionde” :

Not as in “blonde beer”



```
$_='ev  
al("seek\0400  
");foreach(1..3)  
ellhump;my$camel:  
){$_=sprintf("%-6  
t(//);if(defined($  
//);)while(@dromeda  
);if(defined($_=shif  
amelhump+=1<<$CAMEL;  
(@dromedary1))&&/\5/){  
=1<<$CAMEL;}$CAMEL--;if(  
--;if(defined($_=shif  
AS/){$camel1hump+=1<<$CAME  
split(/\n/, $camel);foreach(@  
J\047\134)L^7FX")[$camel1h  
F7\173\175\047\061\062\063\  
JL7F\175\173\047';$_=reverse;  
@camel1hump){chomp;$camel=$_;y  
12345678/JL7F\175\173\0 47'/:  
n");s/\s*/g;eval: ( );s  
/;$_=<DATA>;s/\s*/g:( \124  
\"/)/. {4}/g: DATA__ 41\0  
145\040\157\1 46\040\1 141  
5\1 54\040\1 51\155\ 3\  
156 \040\141 \163\16 3\  
\16 4\151\1 57\156  
1 50\040\ 120\1  
15 1\163\ 040\14  
41\144 \145\  
153\04 0\157  
7\047\ 122\1  
54\171 \040  
101\16 3\16  
43\15 1\14  
145\163 \054  
1\10 \111\156\14 3\056  
1040 125\163\145\14 4\040\  
167\1 51\164\1 50\0 40\160\  
145\162 \155\151  
\163\163 \151\1  
57\156\056
```

Parleremo di:

errate percezioni

delirio e arte

caratteristiche

perl ob. coder

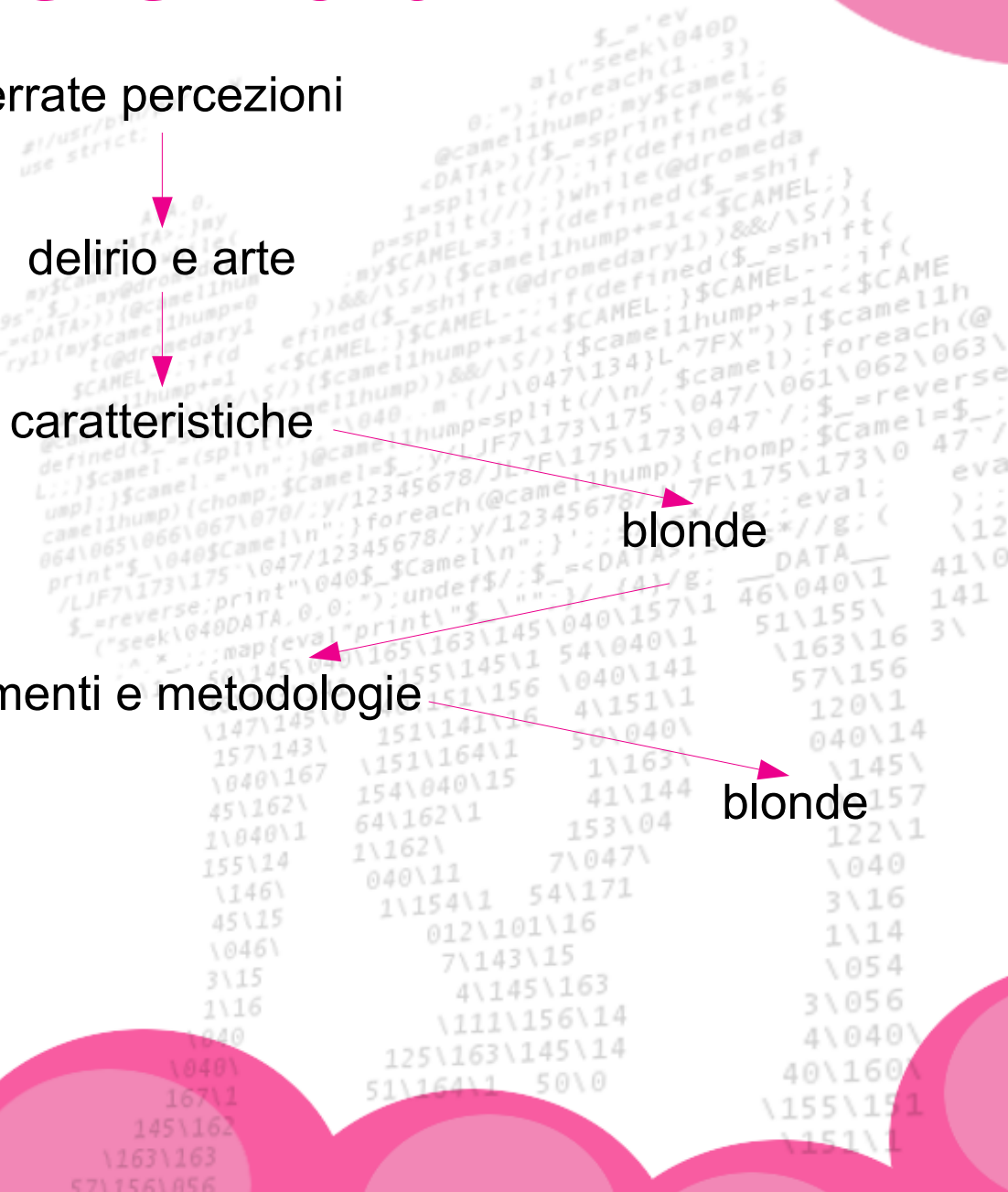
blonde

strumenti e metodologie

perl ob. coder

blonde

offuschiamo



Linux Day

2010

L'incomincio...



```
$_='ev  
al("seek(0400  
0:");foreach(1..3)  
{hump;my$camel:  
($_=sprintf("%-6  
/"));if(defined($  
);)while(@dromeda  
f(defined($_=shif  
l'hump+=1<<$CAMEL:}  
dromedary1))&&/\$/){  
if(defined($_=shif  
l'hump+=1<<$CAMEL  
/)}{$camel1hump+=1<<$CAME  
l'hump)}foreach(@  
047\134)L^7FX")  
l't(/\n/, $camel):  
7F\175\173\047'  
amel1hump){chomp;  
345678/JL7F\175\173\0  
);';s/\s*///g;eval;  
$_=<DATA>;s/\s*///g;  
"/.}{4}/g: DATA  
5\040\157\1 46\040\1 41\0  
1 54\040\1 51\155\ 141  
6 \040\141 \163\16 3\  
6 4\151\1 57\156  
50\040\ 120\1  
1\163\ 040\14  
41\144 \145\  
153\04 0\157  
7\047\ 122\1  
54\171 \040  
012\10\16 1\14  
7\143\15 \054  
4\145\163 3\056  
1\16 \111\156\14 4\040\  
1040 125\163\145\14 40\160\  
167\1 51\164\1 50\0 \155\151  
145\162 \151\  
\163\163  
57\156\056
```


Definizione

- **offuscare** [of-fu-scà-re] v. (offusco, offuschi ecc.) - v.tr. [sogg-v-arg] oscurare qualcosa, renderlo fosco, buio; **privare qualcosa di chiarezza.**



- **offuscare codice:** [ha-ck-th-e-wo-rd] (C, perl, python, ...) - si intende il **rendere il codice sorgente di piu' difficile comprensione**, una **pratica che e' sconsigliabile.**

digressione storica

- Il primo esempio di codice offuscato e' rappresentato dal codice della shell Bourne (**Steve Bourne** in un intorno degli anni '70).
- Bourne decise di utilizzare alcune `#define` del preprocessore C per rendere il suo codice sorgente simile al linguaggio **Algol-68** con il quale aveva maggiore familiarità.



THE
BOURNE
IDENTITY

```
/usr/src/cmd/sh/mac.h:
```

```
#define IF      if(  
#define THEN   ){  
#define ELSE   } else {  
#define ELIF   } else if (  
#define FI     ;}  
  
#define BEGIN  {  
#define END    }  
#define SWITCH switch(  
#define IN     ){  
#define ENDSW }  
#define FOR    for(  
#define WHILE  while(  
#define DO     ){  
#define OD     ;}  
#define REP    do{  
#define PER    }while(  
#define DONE   );  
#define LOOP   for(;;){  
#define POOL   }
```

Definizione

- The **blonde stereotype**: the stereotypical perception of blond-haired women: she is often perceived as making little use of intelligence, as a *"woman who relied rather on her looks rather than on intelligence"*.
- The notion of "dumb blond" has been a topic of academic research reported in scholarly articles and university symposia, which tend to confirm that many people hold to the **perception** that light-haired women are less intelligent than women with dark hair.

Percepito ma...

- Dalle definizioni di cui prima:
 - Blonde “percepita” come diversamente intelligente.
 - Codice Offuscato “percepito” come diversamente consigliabile.
- Ma chi percepisce chi?!?



Linux Day

2010

Pauro dell'incompreso!

Mutt Research Institute



```
#!/usr/bin/perl -w
strict;
al($_, $, '=ev
0:'); foreach(1..3)
@camellhump; my$camel:
)($_, $, '=sprintf("%-6
)($_, $, '=if(defined($
=shif
$CAMEL; )
&&/\S/){
($_, $, '=shift(
CAMEL--; if(
homp+=1<<$CAME
"X")) {$camellh
mel); foreach(@
7/\061\062\063\
47'/:$_=reverse;
homp;$Camel=$_:y
\175\173\0 47'/:
/g;eval: eva
/\s*//g:( );;s
__DATA__
46\040\1 41\0
51\155\ 141
\163\16 3\
57\156
120\1
040\14
\145\
0\157
122\1
\040
3\16
1\14
\054
3\056
4\040\
40\160\
\155\151
\151\1
```

Analisi e Analogie

- Cosa accomuna:



```
#!/usr/bin/perl -w  
use strict;  
  
ATA.O.  
(<DATA>:)my  
my$camel :while(  
$.).ny@dromedary  
@camelhump  
hump=0
```

```
print "Just Another Perl Hacker";
```

?

Analisi e Analogie

- Cosa accomuna:



```
#!/bin/perl
```

```
print "Just Another Perl Hacker";
```

R: attività ritenute NORMALI

Analisi e Analogie

- Cosa accomuna:



```
#!/usr/bin/perl -w  
use strict;
```

```
$_ = "wftedskaebjgdpjgidbsmnjg";  
tr/a-z/oh, turtleneck Phrase Jar!/  
print;
```

?

Analisi e Analogie

- Cosa accomuna:



```
#!/usr/bin/perl -w  
use strict;
```

```
$_ = "wftedskaebjgdpjgidbsmnjg";  
tr/a-z/oh, turtleneck Phrase Jar!/  
print;
```

- stesse identiche operazioni di prima
- qualcuno ha la **percezione** che qualcosa non va in base ai propri **pregiudizi** sul contesto.

stereotypes and prejudices

- Imparare ad apprezzare l'**estetica** di ciò che ci circonda
- Ricominciare a giocare con una sana **logica stringente**
- Non attenersi mai troppo al contesto
- Rivisitare le “regole” in modo più ampio, generico et **fantasioso**
- Farsi offrire una (o più) birre...



Linux Day

2010

examples of stereotypes and prejudices

- il Perl è difficile
- il Python non si offusca

ahahah



01_python-smile.py


```
ddk@hell:~/2010-linuxday/presentazione/code$ cat 01_python-smile.py
```

```
"""
```

```
anche in Python...
```

```
si... puo'... fare!!!
```

```
"""
```

```
        """o;import cosi++*
            ;;from SMILE import ELIMS;*
            ;;map(""";S=5*10+6;SS=chr(S);""o          o$  $$  o$
o $ oo      def S"";SS=chr(S.__lshift__(1));""")  $$ $$ $$o$
oo $ $ "$    #vec1=[]$$ $vec2=Cosi$$$  "";Ss="";""  $p rint$
"SSS="p"$    print "Co"      "si_Presid"    $ente";S=So  "HELO";$
S=TRUE$     "";SS;""$      print Cosi      $SSS+SSS="NoN"*"ESiSte"
$set(['p', 'uu', 'p']);$$  $S+SSS+SSS+S$  S="";SSS="";"";""*SI$
"ot(""";SSSS=SS*2+chr(99-2);SSSSS=SSSS;S=(4*2*(2^3)-1^17)*5-11;"";
nov  o$$[pup1[i]*pup2[i] for i in range(len(pup1))];SS="  ";#$o
oS$"  "pu")"";SSS=chr(S+13)+chr(S+18);""SS="mutt";print"funziona"
#$$  #"";print",;""mutt+funziona!";dict(mutt=9,funziona=4,Cosi=8)
print "";S+S*2;print",;"" sed(xrange(1,2,3,Stella))=42;;map(tux,S);
;print SSSS+$  S=lambda S:S$"";SSSSS=SSS+SSSS;""  print""
    map(      (2010.split()+S););print.split      SS+S
    "cmd=      set(getout.input.A.G.O.W.      );$
    var=      "$$"";def S(S):      o=GS
    S=S_u      SS+S"
    ;print      #collapse(p).pa      .join
    import S      #print SSSS_o      print""
    "object=""";SSSSS=SSSSS+Ss;""
    ;;getattr($_);$_.join;$""
    "";print SSSSS
    #join()+SSS*$
    #print++;$$$
    #true=false
    #LD2010""  A.G.O.W. style
```

```
ddk@hell:~/2010-linuxday/presentazione/code$ cat 01_python-smile.py
```

```
"""
```

```
anche in Python...
```

```
si... puo'... fare!!!
```

```
"""
```

```
        """o;import cosi++*
                ;;from SMILE import ELIMS;*
                ;;map(""";S=5*10+6;SS=chr(S);""o          o$  $$  o$
o $ oo          def S"";SS=chr(S.__lshift__(1));""          $$ $$ $$o$
oo $ $ "$      #vec1=[]$$      $vec2=Cosi$$      "";Ss="";""      $p rint$
"SSS="p"$      print "Co"          "si_Presid"          $ente";S=So          "HEL0";$
S=TRUE$      "";SS;""$          print Cosi          $SSS+SSS="NoN"*"ESiSte"
$set(['p', 'uu', 'p']);$$      $S+SSS+SSS+S$      S="";SSS="";"";""*SI$
"ot(""";SSSS=SS*2+chr(99-2);SSSSS=SSSS;S=(4*2*(2^3)-1^17)*5-11;"";
nov          o$$[pup1[i]*pup2[i] for i in range(len(pup1))];SS="          ";#o
oS$"          "pu)"";SSS=chr(S+13)+chr(S+18);""SS="mutt";print"funziona"
$$$          #"";print"";""mutt+funziona!";dict(mutt=9,funziona=4,Cosi=8)
print "";S+S*2;print"";""sed(xrange(1,2,3,Stella))=42;;map(tux,S);
;print SSS+$      S=lambda S:S$"";SSSSS=SSS+SSSS;""          print""
map(          (2010.split()+S););print.split          SS+S
"cmd=          set(getout.input.A.G.O.W.          );$
var=          "$$";def S(S):          o=GS
S=S_u          SS+S"
;print          #collapse(p).pa          .join
import S          #print SSSS_o          print""
"object=""";SSSSS=SSSSS+Ss;""
;;getattr($_);$_ .join;$""
"";print SSSSS
#join()+SSS*$
#print++;$$$
#true=false
#LD2010""          A.G.O.W. style
```

```
ddk@hell:~/2010-linuxday/presentazione/code$ python 01_python-smile.py
```

```
puppa
```

caratteristiche...

...comuni al buon offuscatore e alle vere bionde:

- visione differente dell' "ambiente"
- indiscussa **fantasia**
- profondo **senso estetico**
- capacita' risolutive **fuori dall'ordinario**
- scopi differenti ma ben precisi
- adattabilita'



Linux Day

2010

Offuscare Codice...

- Non è una tecnica



caratteristiche...

...comuni al buon offuscatore e alle bionde:

- ✓ visione differente dell' "ambiente"
- ✓ fantasia
- ✓ senso estetico
- ✓ capacita' risolutive fuori dall'ordinario
- ✓ scopi differenti ma ben precisi
- ✓ adattabilita'

Linux Day

2010

e il Perl Offuscato?

```
#!/usr/bin/perl -w
use strict;

ATA, 0.
(<DATA>:)my
my$Camel :while(
9s"$_).my@dromedary
-><DATA>))(@camellhump
ry1)(my$camellhump=0
t(@dromedary1
SCAMEL--:;if(d
$camellhump)+1
$camellhump)+1)&&/\S/)($camellhump)
shift(@camellhump)&&/\S/){"\040
m"/J\047\134)L^7FX"}
my$CAMEL=3;if(defined($_)=split(//);)while(@dromeda
<DATA>){$_=sprintf("%-6
1=split(//);)if(defined($
p=split(//);)while(@dromeda
))&&/\S/){$camellhump+1<<$CAMEL;}
fined($_=shift(@dromedary1))&&/\S/){
SCAMEL--:;if(defined($_)=shift(
$CAMEL--:;if(defined($_)=shift(
$camellhump)+1<<$CAMEL)}[$camellh
$camellhump)+1<<$CAMEL)}foreach(@
ump);)camellhump){chomp;$Camel=$_:y
camellhump){chomp;$Camel=$_:y/12345678/JL7F\175\173\0
064\065\066\067\070/.y/12345678/JL7F\175\173\0
print"$_\040$Camel\n";}foreach(@camellhump){chomp;$Camel=$_:y
/LJF7\173\175\1047\12345678/.y/12345678/JL7F\175\173\0
$_=reverse;print"\040$_$Camel\n";}';s/\s*//g;eval(
("seek\040DATA,0,0");undef$/;$_=<DATA>;s/\s*//g;(
: ^ x _:::map{eval"print"$_\n";}/. {4}/g; DATA__
11 50\145\040\163\145\040\157\1 46\040\1 41\0
\147\145\0 155\145\1 54\040\1 51\155\ 141
\147\145\0 40\151\156 1040\141 57\156
157\143\ 151\141\16 4\151\1 120\1
1040\167 \151\164\1 50\040\ 040\14
45\162\ 154\040\15 1\163\ \145\
1\040\1 64\162\1 41\144 0\157
155\14 1\162\ 153\04 122\1
\146\ 040\11 7\047\ \040
45\15 1\154\1 54\171 3\16
1046\ 012\101\16 1\14
3\15 7\143\15 \054
1\16 4\145\163 3\056
1040 \111\156\14 4\040\
1040\ 125\163\145\14 40\160\
167\1 51\164\1 50\0 \155\151
145\162 \163\163 \151\1
\163\163 57\156\056
```

Linux Day

2010

Perl... esempio

02_eureka.pl

```
#!/usr/bin/perl -w
use strict;

ATA,0.
(<DATA>:)my
my$camel :while(
9s"$_).my@dromedary
->"DATA>))(@camellhump
ry1)(my$camellhump=0
t(@dromedary1
SCAMEL--:if(d
$camellhump+1
($camellhump)&&/\S/){$camellhump}
//."1040..m"/J\047\134\L^7FX")
@camellhump=split(/\n/,$camel);foreach(@
camelhump){chomp;$camel=$_;eval(
064\065\066\067\070/:y/12345678/JL7F\175\173\047/:$_=reverse;
print"$_\040$camel\n";}foreach(@camellhump){chomp;$camel=$_;
/LJF7\173\175\1047/12345678/:y/12345678/JL7F\175\173\047/:
$_=reverse;print"\040$_$camel\n";}';s/\s*//g;eval(
("seek\040DATA,0,0");undef$/;$_=<DATA>;s/\s*//g;(
: ^x...:map{eval"print"$_\n";}/.{})/g;
DATA__
46\040\1 41\0
51\155\ 141
\163\16 3\
57\156
120\1
040\14
\145\
0\157
122\1
1\040\1
153\04
\040
040\11 7\047\
3\16
012\101\16
7\143\15
\054
4\145\163
3\056
\111\156\14
4\040\
125\163\145\14
40\160\
51\164\1 50\0
\155\151
\151\1
```


ddk@hell:~/2010-linuxday/presentazione/code\$ cat 02_eureka.pl

!
!
!
!
!

!
!
!
!

chr (\$!

=>! chr ,

```

    ;$_=join(
    q!!=>map{$!=$!&
    0;map{s$![^\!] =s!!
    \x-Y\[@!]!Y=$! =$x
    ;$!+=s[q!=>\x-Y= [@!
    ]$!={$g}@$_;$g=i&& q!!
    +0x3E)}([\!$!=>{!! !$!
    [!$!]}=>[\!$!=>\!$!=>\!$!
    =>[!$!]}=>[\!$!=>\!$!=>
    {$!=>[$!]}=>{!$!=>[
    $!]}=>[\!$!=>{$!
    =>[Y]}=>[\!$!=>
    {$!=>[!$!]}=>
    \!$!]}=>[
    \!$!]))
    print=>
    print=>
    print=>
    print=>
    print=>
    print

```

!
!
!
!

ddk@hell:~/2010-linuxday/presentazione/code\$ cat 02_eureka.pl

```
!
!
!
!
!
i
!
```

. chr (\$!

```
;$_=join(
  q!!=>map{!=$!&
  0;map{s$![^\!] =s!!
  \x-Y\[@!]!Y=$! =$x
  ;$!+=s[q!>=\x-Y= [@!
  ]$!={$g}@$_;$g=i&& q!!
  +0x3E)}([\!$!>{!! !$!
  [!$!]}=>[\!$!>!\!>!\!
  =>[!$!]}=>[\!$!>!\!>
  {$!>[$!]}=>{!$!>[
  $!]}=>[\!$!>{$!
  =>[Y]}=>[\!$!>
  {$!>[!$!]}=>
  \!$!>[
  \!$!]))
  ;
  print=>
  print=>
  print=>
  print=>
  print=>
  print
```

=>! chr ,

ddk@hell:~/2010-linuxday/presentazione/code\$ perl 02_eureka.pl

EUREKA

Linux Day

2010

Perl... esempio

03_camel-code.pl

```
#!/usr/bin/perl -w
use strict;

ATA,0.
(<DATA>:)my
my$Camel :while(
9s"$_).my@dromedary
->"<DATA>))(@camellhump
ry1)(my$camellhump=0
t(@dromedary1
SCAMEL--:;if(d
$camellhump++1
@camellhump)&&/\S/){$camellhump}
defined($_=shift(@camellhump))&&/\S/){
$Camel=(split//,"1040..m"/J\047\134)L^7FX")
foreach(@camellhump){chomp;$Camel=$_;
print"$_\040$Camel\n";}
s/\s*/g;eval(
("seek\040DATA,0,0");undef$/;$_=<DATA>;s/\s*/g;
^x...;map{eval"print"$_\n";}/.{}g;
DATA__
46\040\1 41\0
51\155\ 141
\163\16 3\
57\156
120\1
040\14
\145\
0\157
122\1
1\040\1
153\04
\040
040\11 7\047\
3\16
12\101\16
7\143\15
\054
4\145\163
3\056
\111\156\14
4\040\
125\163\145\14
40\160\
51\164\1 50\0
\155\151
\151\1
```

```

ddk@hell:~/2010-linuxday/presentazione/code$ cat 03_camel-code.pl
#!/usr/bin/perl -w                                # camel code
use strict;

```

```

                                $_='ev
                                al("seek\040D
ATA,0,                                0;");foreach(1..3)
    {<DATA>};my                                @camellhump;my$camel;
my$Camel ;while(                                <DATA>){$_=sprintf("%-6
9s",$_);my@dromedary                                1=split(/);if(defined($
_=<DATA>)){@camellhum                                p=split(/);}while(@dromeda
ry1){my$camellhump=0                                ;my$CAMEL=3;if(defined($_=shif
t(@dromedary1                                ))&&/\S/){$camellhump+=1<<$CAMEL;}
$CAMEL--;if(d                                efinid($_=shift(@dromedary1))&&/\S/){
$camellhump+=1                                <<$CAMEL;}$CAMEL--;if(defined($_=shift(
@camellhump))&&/\S/){$camellhump+=1<<$CAMEL;}$CAMEL--;if(                                defined($_=shift(@camellhump))&&/\S/){$camellhump+=1<<$CAME
L;}}$camel.=split(/, "\040..m`{/J\047\134}L^7FX")[$camellh                                ump];}$camel.="\\n";}@camellhump=split(/\\n/, $camel);foreach(@
camellhump){chomp;$Camel=$_;y/LJF7\173\175`\047/\061\062\063\                                064\065\066\067\070;/y/12345678/JL7F\175\173\047`/;$_=reverse;
print"$_\040$Camel\\n";}foreach(@camellhump){chomp;$Camel=$_;y                                /LJF7\173\175`\047/12345678;/y/12345678/JL7F\175\173\047`/;
$_=reverse;print"\040$_$Camel\\n";}';;s/\s*//g;;eval; eval                                ("seek\040DATA,0,0;");undef$/;$_=<DATA>;s/\s*//g;( );;s
^.*_;;map{eval"print"$_\040";}/. {4}/g; DATA                                \124
\1_ 50\145\040\165\163\145\040\157\1 46\040\1 41\0                                40\143\141 \155\145\1 54\040\1 51\155\ 141
\147\145\0 40\151\156 \040\141 \163\16 3\
157\143\ 151\141\16 4\151\1 57\156
\040\167 \151\164\1 50\040\ 120\1
45\162\ 154\040\15 1\163\ 040\14
1\040\1 64\162\1 41\144 \145\
155\14 1\162\ 153\04 0\157
\146\ 040\11 7\047\ 122\1
45\15 1\154\1 54\171 \040
\046\ 012\101\16 3\16
3\15 7\143\15 1\14
1\16 4\145\163 \054
\040 \111\156\14 3\056
\040\ 125\163\145\14 4\040\
167\1 51\164\1 50\0 40\160\
145\162 \155\151
\163\163 \151\1
57\156\056

```

```

ddk@hell:~/2010-linuxday/presentazione/code$

```


ddk@hell:~/2010-linuxday/presentazione/code\$ perl 03_camel-code.pl

```

      mJXXLm.          JXXLm.          .mJXXL          .mJXXLm
      JXXXXXXXXXL.    JXXLm.          .JXXXXXXXXXL
      {XXXXXXXXXXXXX.  JXXmXXXXm mXXXXmXXXL .XXXXXXXXXXXXX}
      .XXXXXXXXXXXXXL. {XXXXXXXXXXF 7XXXXXXXXXX} .JXXXXXXXXXXXXX.
      JXXXXXXXXXXXXXXXXXXXXL. `XXXXXX. .XXXXXX'.JXXXXXXXXXXXXXXXXXXL
      JXXXXXXXXXXXXXXXXXXXXXXXXXXXXmXXXXXXXX. .XXXXXXXXmXXXXXXXXXXXXXXXXXXXXL
      .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX} {XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
      .XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  7XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      XX'7XXXXXXXXXXXXXXXXXXXXXXXXXXXXF`XX  7XXXXXXXXXXXXXXXXXXXXXXXXXXXXF`XX
      XX {XXXFXXXXXXXXXXXXXXXXXXXXXF'      `7XXXXXXXXXXXXXXXXXXXX7XXX} XX
      `X}-{XXX'7XXXFXXXXXX^XXXXX ` ` '  XXXXX^XXXXX7XXXF`XXX}-{X'
      `XXX' {XXX'XXXXX 7XXXF          7XXXF XXXXX`XXX} `XXX` `
      .XX} {XXF {XXXX} `XXX}          {XXX' {XXXX} 7XX} {XX.
      {XX `XXL `7XX} 7XX}           {XXF {XXF' JXX'  XX}
      `XX `XXL mXXF {XX              XX} 7XXm JXX'  XX'
      XX 7XXXF `XX                  XX' 7XXXF  XX
      XX. JXXX. 7X.                  .XF .XXXXL .XX
      {XXL 7XF7XXX. {XX              XX} .XXXF7XF JXX}
      `XXX' `XXXm                    mXXX' `XXX'
      ~~~~~

```

```

      .mJXXLm          mJXXLm.
      .mJXXL          JXXXXXXXXXL          JXXLm.
      mXXXXmXXXL      .XXXXXXXXXXXXX}      {XXXXXXXXXXXXX.  JXXmXXXXm
      7XXXXXXXXXX} .JXXXXXXXXXXXXX.          .XXXXXXXXXXXXXL. {XXXXXXXXXXF
      .XXXXXX'.JXXXXXXXXXXXXXXXXXXL        JXXXXXXXXXXXXXXXXXXXXL.`XXXXXX.
      .XXXXXXXXmXXXXXXXXXXXXXXXXXXXXL      JXXXXXXXXXXXXXXXXXXXXmXXXXXX.
      {XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.      .XXXXXXXXXXXXXXXXXXXXX}
      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.  .XXXXXXXXXXXXXXXXXXXXX
      7XXXXXXXXXXXXXXXXXXXXX             XXXXXXXXXXXXXXXXXXXXXXXX
      7XXXXXXXXXXXXXXXXXXXXX`XX          XX'7XXXXXXXXXXXXXXXXXXXXX
      `7XXXXXXXXXXXX7XXX} XX           XX {XXXFXXXXXXXXXXXXXF'
      `X}-{XXX'7XXXFXXXXXX^XXXXX ` ` '  XXXXX^XXXXX7XXXF`XXX}-{X'
      7XXXF XXXXX`XXX} `XXX` `
      {XXX' {XXXX} 7XX} {XX.
      {XXF {XXF' JXX'  XX}
      XX} 7XXm JXX'  XX'
      XX' 7XXXF  XX
      .XF .XXXXL .XX
      XX} .XXXF7XF JXX}
      mXXX' `XXX'
      ~~~~~

```

The use of a camel image in association with Perl is a trademark of O'Reilly & Associates, Inc. Used with permission.ddk@hell:~/2010-linuxday/presentazione/code\$

f3802@optiplex:~/2010-linuxday/presentazione/code\$ cat 04_camel-code-modding.pl

```
#!/usr/bin/perl -w # camel code
use strict;
```

```
$ = 'eval("seek\040DATA,0,0;");foreach(1..3){<DATA>};
my@camellhump;my$camel;my$Camel;while(<DATA>){$_=spr
intf("%-69s",$_);my@dromedary1=split(/);if(defined(
$_=<DATA>)){@camellhump=split(/);}while(@dromedary1
){my$cam ellhump=
0;my$CAM EL=3;if(
defined( $_=shif
t(@dromed ary1))&&
(/S/){$c amel1hum p+=1<<$C
AMEL--;i f(define d($_=shi
ft(@drom edary1))
&&/\S/){ $ump+=1<<
$CAMEL--
;if(defi ned($ =s
hift(@ca mellhump
))&&/\S/
){$camel <<$CAMEL
lhump+=1 =shift(@
;$CAMEL --;if(de fined($_
camellhu mp))&&/\ S/){$cam ellhump+
=1<<$CAM EL;;}$ca mel.=
(J\047\ X"))[$$ca l."n";
}]}$came l."n";
}@camellhump=split(/\n/, $camel);foreach(@camellhump)
{chomp;$Camel=$_;y/LJF7\173\175` \047/\061\062\063\06
4\065\066\067\070;/y/12345678/JL7F\175\173\047` /;$_=
reverse;print"$_\040$Camel\n";}foreach(@camellhump){
chomp;$Camel=$_;y
/LJF7\173\175` \04
7/12345678;/y/123
45678/JL7F\175\17
3\0 47` /;$_=rever
se;print"\040$_$Camel\n";}';;s/\s*///g;eval;eval("se
ek\040DATA,0,0;");undef$/$$_=<DATA>;s/\s*///g;();;s;^
.*_;;map{eval"print"$ \"";}/. {4}/g; DATA__\124\1
50\145\040\165\163\145\040\157\
146\040\141\040\143\141\155
\145\154\040\151\155\14
1\147\145\040\151\15
6\040\141\163\163\15
7\143\151\141\164\15
1\157\156\040\167\
151\164\150\040\12
0\145\162\154\040\
151\163\040\141\
040\164\162\141\
144\145\155\141\
162\153\040\157\
146\040\117\047\122\1
45\151\154\154\171\040
\046\012 \101\163
\163\157 \143\151
\141\164\ 145\163\0
54\040\111 \156\143\0
56\040\125\ 163\145\144
\040\167\15 1\164\150\0
40\160\145\ 162\155\151
\163\163\151 \157\156\056
```


Linux Day

2010

caratteristiche...

WYSINWYG



camel code

```
#!/usr/bin/perl -w
strict;

$_='ev
al("seek\0400
0:");foreach(1..3)
@camellhump;my$camel;
)($_=sprintf("%-6
");if(defined($
$_=shif
<$CAMEL;
)&&/\s/){
d($_=shift(
$CAMEL--);if(
hump+=1<<$CAME
"FX")){ $camellh
camel);foreach(@
047/\061\062\063\
047'/:$_=reverse;
chomp;$Camel=$_;y
\F\175\173\0 47'/:
//g;eval;
s/\s*//g;(
; DATA
1 46\040\1 41\0
51\155\ 141
\163\16 3\
57\156
120\1
040\14
\145\
0\157
122\1
\040
3\16
1\14
\054
3\056
4\040\
40\160\
\155\151
\151\1
\046\
7\143\15
3\15
4\145\163
1\16
\111\156\14
\040
125\163\145\14
51\164\1 50\0
\163\163
57\156\056
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ cat tf.c
```

```
#include <stdio.h>
#define sub main
#define True printf
#define print /* False */
#define $True /*
;;$True = "True\n";;;;;;$False = */ ;;
;;print $True;sub(){True ("False\n");;}
#define me A.G.O.W.
#define End .
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ gcc tf.c -o tf
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ cat tf.c
```

```
#include <stdio.h>
#define sub main
#define True printf
#define print /* False */
#define $True /*
;;$True = "True\n";;;;;;$False = */ ;;
;;print $True;sub(){True ("False\n");;}
#define me A.G.O.W.
#define End .
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ gcc tf.c -o tf
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ ./tf
```

```
False
```



```
f3802@optiplex:~/2010-linuxday/presentazione/code$ cat tf.c
```

```
#include <stdio.h>
#define sub main
#define True printf
#define print /* False */
#define $True /*
;;$True = "True\n";;;;;;$False = */ ;;
;;print $True;sub(){True ("False\n");;}
#define me A.G.O.W.
#define End .
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ gcc tf.c -o tf
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ ./tf
```

```
False
```

```
f3802@optiplex:~/2010-linuxday/presentazione/code$ perl tf.c
```

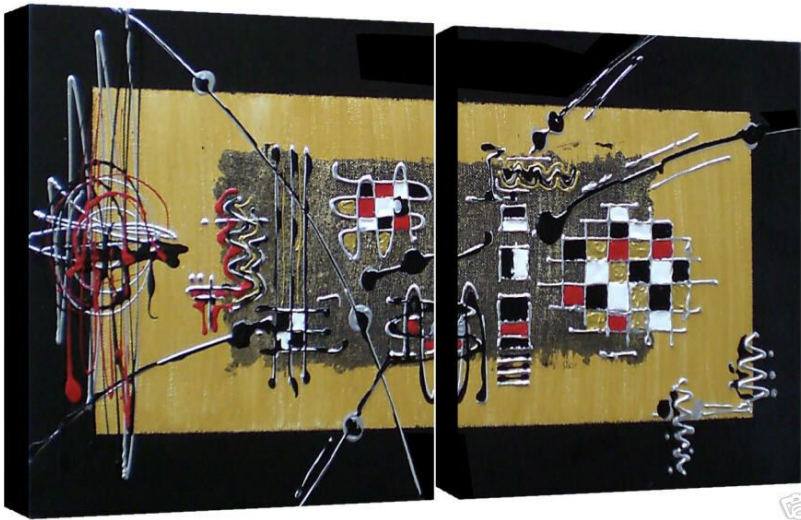
```
True
```

Linux Day

2010

Non c'e' Arte senza Tecnica

- Sia per essere "bionde"
- Sia per codare offuscato



```
#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>:)my
my$CAMEL :while(
90%...)@romedary
-><DATA>@romedary
ry1)(my@romedary
SCAMEL--:;fr
$camelhump++1 <<$CAMEL
@camelhump)&&/\$/){$camelhump
L:)}$camel=(split(//,
camelhump){chomp;$CAMEL=}
54\145\140\165\163\145\140\157\1
11 40\143\141 155\145\1 54\140\1
\147\145\0 40\151\156 1040\141
257\143\ 151\141\16 4\151\1
1040\167 \151\164\1 50\1040\
45\162\ 64\162\1 41\163\
1\1040\1 1\162\ 153\104
155\14 040\11 7\1047\
\146\ 1\154\1 54\171
45\15 012\101\16
\1046\ 7\143\15
3\15 4\145\163
1\16 \111\156\14
1040 125\163\145\14
1040\ 51\164\1 50\0
167\1
145\162
\163\163
57\156\156
```

Linux Day

2010

Blonde/Rules



```
#!/usr/bin/perl -w  
strict;
```

```
$_='ev  
al("seek\0400  
0:"); foreach(1..3)  
{my$hump;my$camel;  
$camel=sprintf("%-6  
s",defined($  
e{@dromeda  
ned($_)=shif  
+=1<<$CAMEL;)  
ary1))&&/\S/){  
fined($_=shift(  
EL:)}$CAMEL--;if(  
ellhump+=1<<$CAME  
L^7FX")) {$camelh  
/, $camel); foreach(@  
5'\047\1061\062\063\  
173\047';$_=reverse;  
mp){chomp;$Camel=$_;  
/JL7F\175\173\0 47'/;  
/\s*//g;eval: eva  
TA>;s/\s*//g;( );;s  
4)/g: DATA__ \124  
157\1 46\040\1 41\0  
40\1 51\155\ 141  
\141 \163\16 3\  
51\1 57\156  
040\ 120\1  
163\ 040\14  
144 \145\  
04 0\157  
\ 122\1  
 \040  
 3\16  
45\15 1\154\1  
 \046\ 012\101\16  
 3\15 7\143\15  
 1\16 4\145\163  
 \040 \111\156\14  
 \163\ 125\163\145\14  
 167\1 51\164\1 50\0  
 145\162  
 \163\163  
 57\156\056
```

camel code

Linux Day

2010

Tools



```
#!/usr/bin/perl -w  
use strict;
```

```
$_='ev  
al("seek\0400  
0:"); foreach(1..3)  
@camellhump;my$camel;  
<DATA>){$_=sprintf("%-6  
11t//");if(defined($  
while(@dromeda  
ined($_=shif  
P+=1<<$CAMEL;)  
dary1))&&/\S/){  
efined($_=shif  
MEL;)$CAMEL--;if(  
amellhump+=1<<$CAME  
4)L^7FX")[$camellh  
n/,$camel);foreach(@  
75'\047'\061\062\063\  
\173\047';$_=reverse;  
ump){chomp;$Camel=$_/y  
8/JL7F\175\173\0 47'/;  
s/\s*//g;eval: eva  
ATA>;s/\s*//g;( );;s  
{4}/g: DATA__ 41\0  
\157\1 46\040\1 141  
040\1 51\155\ 141  
0\141 \163\16 3\  
151\1 57\156  
040\ 120\1  
163\ 040\14  
\144 \145\  
\04 0\157  
104 122\1  
155\14 1\162\ 7\047\  
040\11 54\171  
\146\ 1\154\1 3\16  
45\15 012\101\16 1\14  
1046\ 7\143\15 \054  
3\15 4\145\163 3\056  
1\16 \111\156\14 4\040\  
040 125\163\145\14 40\160\  
167\1 51\164\1 50\0 155\151  
145\162 \151\1  
\163\163  
57\156\056
```

camel code

Linux Day

2010

Soluzioni Semplici

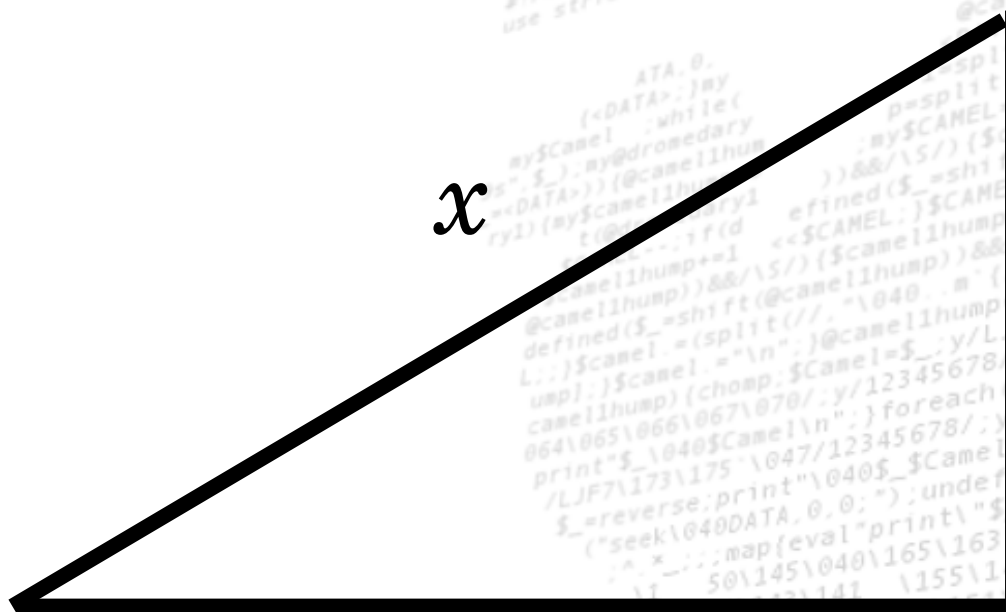
quanto geniali



Linux Day

2010

Trovare x



x

3

4

```
#!/usr/bin/perl -w
use strict;

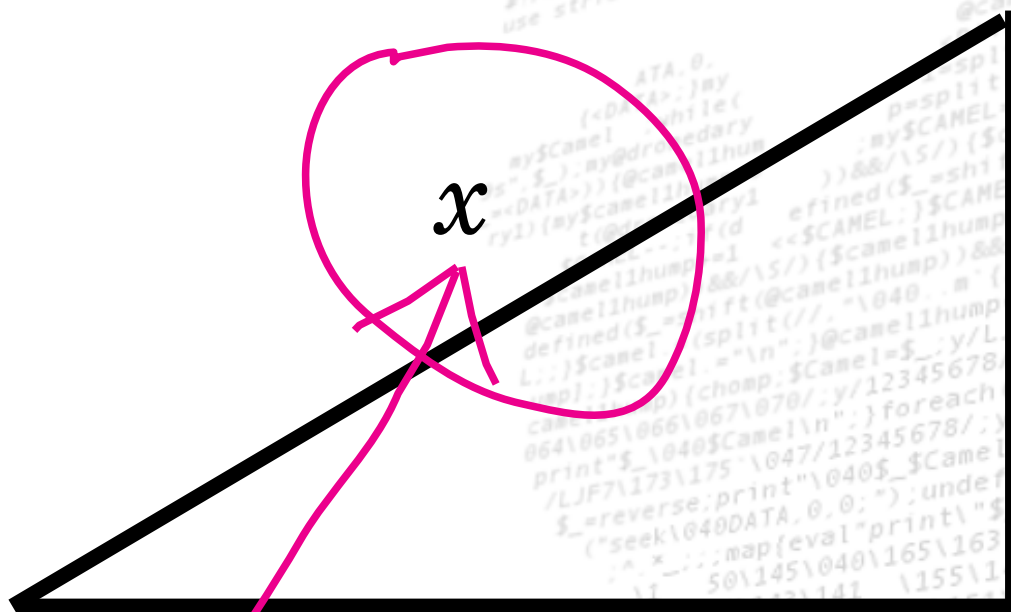
ATA, 0,
(<DATA>:)my
my$Camel :while(
s"$_".my@dromedary
=<DATA>))@camel1hump
ry1)(my$camel1hump
t(@camel1hump
camel1hump++1 <<$CAMEL:$CAMEL
@camel1hump)&&/\S/){$camel1hump
defined($_=shift(@camel1hump)
L:)}$camel.=("n");@camel1hump
ump);$camel={chomp;$CAMEL=$_;y/L
camel1hump}{chomp;$CAMEL=$_;y/L
064\065\066\067\070/:y/12345678
print"$_\040$CAMEL\n";}foreach
/LJF7\173\175"\047/12345678/:
$_=reverse;print"\040$_$CAMEL
("seek\040DATA,0,0:");under
^x...:map{eval"print"$
50\145\040\165\163
51\145\040\155\1
\147\145\0
151\141\16
257\143\
\040\167
\251\164\1
45\162\
154\040\15
1\040\1
64\162\1
1\162\
153\04
155\14
040\11
7\047\
\146\
1\154\1 54\171
45\15
012\101\16
\046\
7\143\15
3\15
4\145\163
1\16
\111\156\14
\040
125\163\145\14
\040\
51\164\1 50\0
167\1
145\162
\163\163
57\156\056
```

camel code

Linux Day

2010

Trovare x

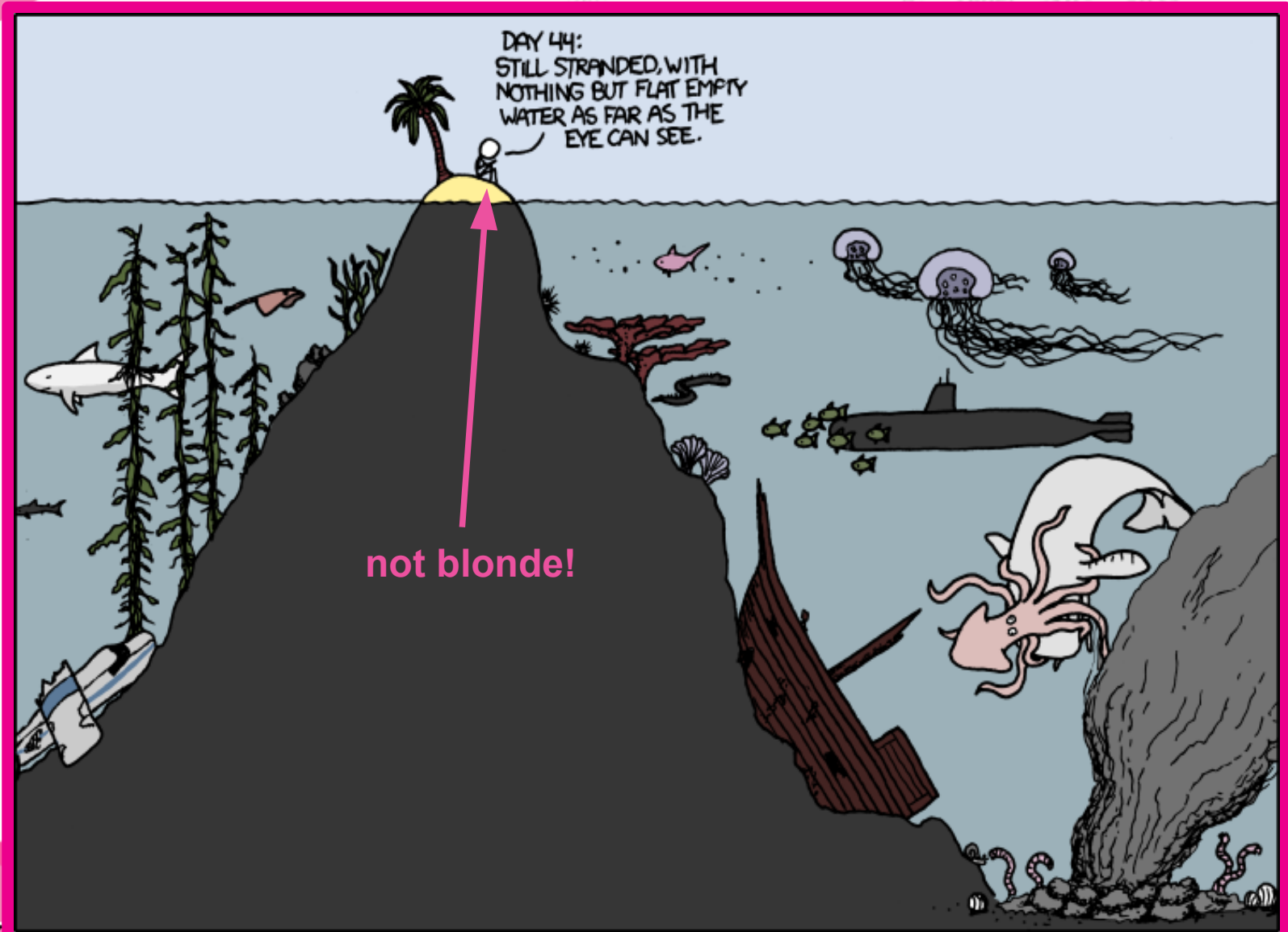


eccola

Linux Day

2010

spirito d'osservazione



camel code

```
$_ = 'ev  
al ("seek\0400  
"); foreach (1..3)  
    print my$camel;  
    print f("%-6  
45
```

```
ME  
11h  
ch (@  
1063\  
verse;  
l=$_  
47"/;  
eva  
);:s  
124  
41\0  
141  
3\  
4  
7  
1
```

Linux Day

2010

Spirito d'osservazione

dall'osservazione di:

$$\lim_{x \rightarrow 8} \frac{1}{x - 8} = \infty$$

```

#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>:)my
@camel :while(
my@camel :my@dromedary
))(@camelhump
my$camelhump=0
t(@dromedary1
SCAMEL--:;if(d
$camelhump++1
@camelhump)&&/\S/)
defined($_=shift(@camelhump)&&/\S/){
L:)}$camel.=("\n");@camelhump=split(/\n/,$camel);foreach(@
camelhump){chomp;$camel=$camel.$camel;foreach(@
064\065\066\067\070/:y/12345678/JL7F\175\173\047/:;$_=reverse;
print"$_\040$Camel\n";}foreach(@camelhump){chomp;$camel=$_:y
/LJF7\173\175\1047/12345678/:;s/\s*//g;eval(
reverse;print"\040$_Camel\n";}undef$/;$_=<DATA>;s/\s*//g;(
ATA.0.0:);undef$/;$_=<DATA>;s/\s*//g;(
50\145\040\165\163\145\040\157\1
40\143\141 155\145\1 54\040\1
\147\145\0 40\151\156 1040\141
257\143\ 151\141\16 4\151\1
1040\167 \125\164\1 50\040\
45\162\ 154\040\15 1\163\
1\040\1 64\162\1 41\144
155\14 1\162\ 153\04
\146\ 040\11 7\047\
45\15 1\154\1 54\171
1046\ 012\101\16
3\15 7\143\15
1\16 4\145\163
1040 \111\156\14
167\1 125\163\145\14
145\162 51\164\1 50\0
\163\163 \40\160\
57\156\056 \155\151
\151\1

```

camel code

Linux Day

2010

manipolazione di significanti, significati e sintassi



Linux Day

2010

manipolazione di significanti, significati e sintassi

$$\frac{1}{n} \sin x =$$

```
#!/usr/bin/perl -w
use strict;

ATA, 0.
(<DATA>:)my
my$camel :while(
9s"$_).my@dromedary
->"<DATA>))(@camel|hump
ry1)(my$camel|hump=0
t(@dromedary1
SCAMEL--:;if(d
$camel|hump++1
@camel|hump)&&/\S/){$camel|hump}
defined($_=shift(@camel|hump))&&/\S/){
$camel.=split(/,/,"1040..m"/);
}foreach(@camel|hump){chomp;$camel=$_.y
("seek1040DATA,0,0");undef$/;$/={}/g;
^x...;map{eval"print\"$_\"";}$.<DATA>;s/\s*//g;
50\145\1040\165\163\145\1040\157\1
40\143\141 \155\145\1 54\1040\1
\147\145\10 40\151\156 \1040\141
257\143\ 151\141\16 4\151\11
\1040\167 \125\164\1 50\1040\
45\162\ 64\162\1 41\163\
1\1040\1 1\162\ 153\104
155\14 \040\11 7\1047\
\146\ 1\154\1 54\171
45\15 \012\101\16
\1046\ 7\143\15
3\15 4\145\163
1\16 \111\156\14
1040 125\163\145\14
\1040\ 51\164\1 50\10
167\1
145\162
\163\163
57\156\1056
```


Linux Day

2010

manipolazione di significanti, significati e sintassi

$$\frac{1}{n} \sin x = \sin x$$

```
#!/usr/bin/perl -w
use strict;


ATA,0.
(<DATA>:)my
my$camel :while(
9s"$_).my@dromedary
->"<DATA>))(@camelhump
ry1)(my$camelhump=0
t(@dromedary1
SCAMEL--:;if(d
$camelhump++1
@camelhump)&&/\S/){$camelhump}
defined($_=shift(@camelhump))&&/\S/){
$camel=(split//,"1040..m"/J\047\134)L^7FX");
foreach(@camelhump){chomp;$camel=$_:y
print"$camel\n";}
$_=reverse;print"$_\n";}/.{})/g:
("seek1040DATA,0,0");undef$/;$_=<DATA>;s/\s*//g:
: ^x...;map{eval"print"$_\n";}$_;
50\145\1040\165\163\145\1040\157\1
40\143\141 155\145\1 54\1040\1
\147\145\0 40\151\156 1040\141
257\143\ 151\141\16 4\151\1
1040\167 1251\164\1 50\1040\
45\162\ 64\162\1 1\163\
1\1040\1 1\162\ 153\104
155\14 040\11 7\1047\
\146\ 1\154\1 54\171
45\15 012\101\16
1046\ 7\143\15
3\15 4\145\163
1\16 \111\156\14
1040 125\163\145\14
1040\ 51\164\1 50\0
167\1
145\162
\163\163
57\156\056
```


Linux Day

2010

manipolazione di significanti, significati e sintassi

test your "blondness"

$$\sqrt{2} = 2$$


```
#!/usr/bin/perl -w
use strict;

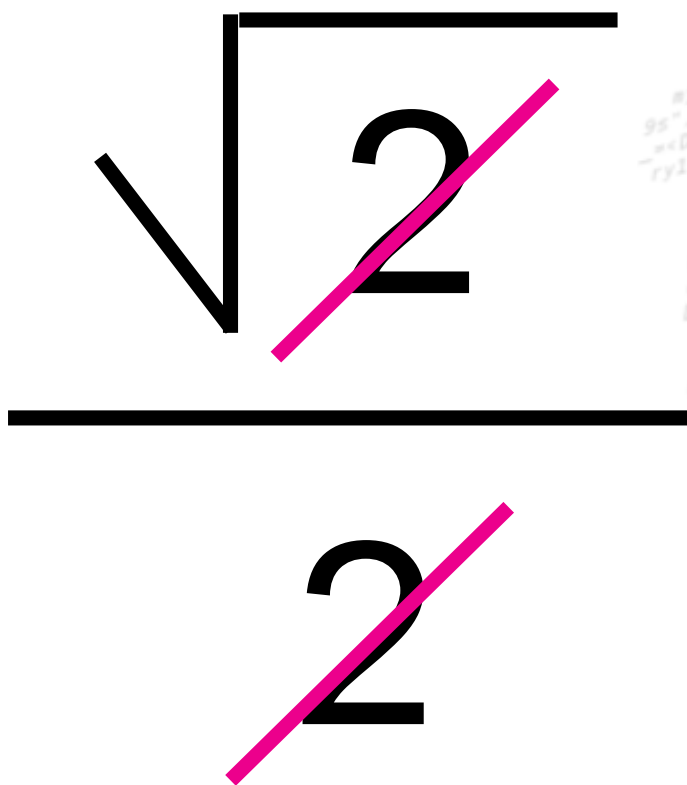
ATA, 0.
(<DATA>);my
my$CAMEL :while(
9s"$_).my@dromedary
->"<DATA>))(@cAMELhump=0
ry1)(my$cAMELhump=0
t(@dromedary1
SCAMEL--:;if(d
$cAMELhump+1
@cAMELhump)&&/\S/){$cAMELhump+1<<$CAMEL;}
defined($_=shift(@cAMELhump)&&/\S/){
L:)}$cAMEL="";@cAMELhump=$_
ump);$cAMEL{chomp;$CAMEL=$_;
cAMELhump}{chomp;$CAMEL=$_;
064\065\066\067\070/:y/12345
print"$cAMEL\n";}for
/LJF7\173\175\047\123456789\
$_=reverse;
("seek\0400ATA,0,0");
:^\x...;map{eval"print
50\145\040\165
11
40\143\141
\147\145\0
157\143\
1040\167
45\162\
1\040\1
155\14
\146\
45\15
\046\
3\15
1\16
1040
1040\
167\1
145\162
\163\163
57\156\056
155\
\151\1
46\040\1
51\155\
\163\16
57\156
120\1
040\14
\145\
0\157
122\1
\040
3\16
1\14
\054
3\056
4\040\
40\160\
\155\151
\151\1
```

Linux Day

2010

Significanti, Significati e Sintassi

yep! U r !!



```

#!/usr/bin/perl -w
use strict;

ATA,0.
(<DATA>:)my
my$Camel :while(
9s"$_).my@dromedary
-><DATA>))(@camelhump
ry1)(my$camelhump=0
t(@dromedary1
SCAMEL--:;if(d
$camelhump++1
@camelhump)&&/\S/)
defined($_=shift(@camelhump))&&/\S/){
L:)}$camel="n":)@camelhump
ump);}$camel=$_:y/
camelhump){chomp;$Camel=$_:y/
064\065\066\067\070/:y/1234567
print"$camel\n";}foreach
/LJF7\173\175\1047/12345678/
{"seek\0400ATA,0,0";}und
$_="";}{4}/g:
50\145\040\165\155\145\154\040\1
40\143\141\155\156\1040\141
\147\145\040\151\141\164\151\1
257\143\151\164\150\040\
1040\167\154\040\15\1\163\
45\162\64\162\141\144
1\040\11\162\153\04
155\14\040\117\047\
\146\040\1154\154\171
45\15\012\101\16
1046\7\143\15
3\154\145\163
1\16\111\156\14
1040125\163\145\14
1040\167\151\164\150\0
145\162\163\163
57\156\056

```

Linux Day

2010

estetica

idee geniali

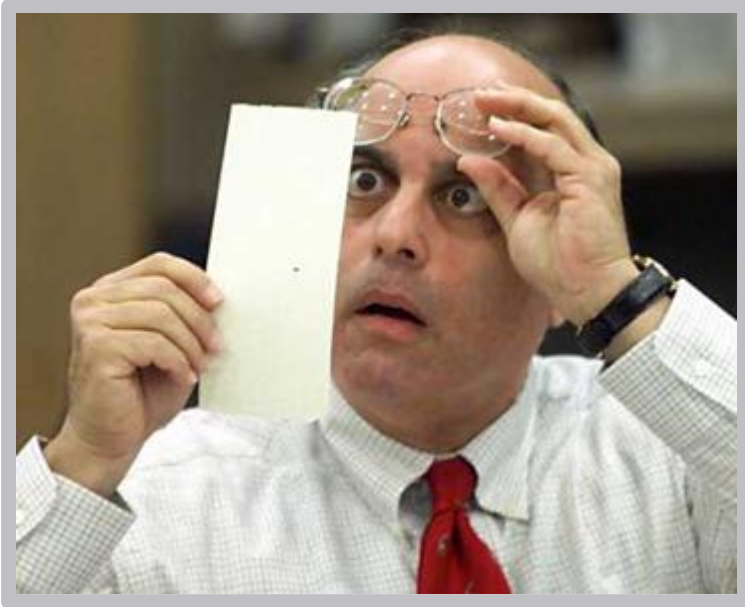


```
$_='ev  
al("seek\040D  
reach(1..3)  
p:my$camel:  
printf("%-6  
if(defined($  
while(@dromeda  
defined($_=shif  
ump+=1<<$CAMEL;  
edary1))&&/\$/){  
(defined($_=shif  
CAMEL:)$CAMEL--:if(  
$camel1hump+=1<<$CAME  
134)L^7FX") [$camel1h  
/\n/, $camel); foreach(@  
\175\047\061\062\063\  
75\173\047\175\173\0  
1hump){chomp;$Camel=$_:y  
678/JL7F\175\173\0 47\/  
:s/\s*//g;eval: eva  
<DATA>;s/\s*//g:( );;s  
/{4}/g: DATA__ \124  
40\157\1 46\040\1 41\0  
1\040\1 51\155\ 141  
040\141 \163\16 3\  
\151\1 57\156  
0\040\ 120\1  
1\163\ 040\14  
41\144 \145\  
53\04 0\157  
047\ 122\1  
171 \040  
5 3\16  
 1\14  
 \054  
53 3\056  
 \111\156\14 4\040\  
125\163\145\14 40\160\  
51\164\1 50\0 \155\151  
 151\1
```

Estetica!

- Expand:

$$(a+b)^n$$



```
#!/usr/bin/perl -w
strict;

ATA.0.
(<DATA>);my
my$Camel :while(
9s".$_).my@dromedary
->"<DATA>))(@camellhump
ry1)(my$camellhump=0
t(@dromedary1
SCAMEL--:;if(d
$camellhump+1
@camellhump)&&/\S/){$camellhump+1<<$CAMEL;}
defined($_=shift(@camellhump))&&/\S/){
L:)}$camel=(split//,"1040..m"/\047\134)L^7FX");
camellhump){chomp;$Camel=$Camel+1<<$CAMEL;foreach(@
064\065\066\067\070/:y/12345678/JL7F\175\173\047/:;$_=reverse;
print"$\040$Camel\n";}foreach(@camellhump){chomp;$Camel=$_:y
/LJF7\173\175\1047/12345678/:;s/\s*///g;eval:
$_=reverse;print"\040$_$Camel\n";};s/\s*///g:(
("seek\040DATA,0,0");undef$/;$_=<DATA>;s/\s*///g:
: ^ x :;map{eval"print"$\1";}/. {4}/g: DATA
11 50\145\040\165\163\145\040\157\1 46\040\1 41\0
40\143\141 155\145\1 54\040\1 51\155\ 141
\147\145\0 40\151\156 1040\141 57\156
257\143\ 151\141\16 4\151\1 120\1
1040\167 \151\164\1 50\040\ 040\14
45\162\ 154\040\15 1\163\ \145\
1\040\1 64\162\1 41\144 0\157
155\14 1\162\ 153\04 122\1
\146\ 040\11 7\047\ \040
45\15 1\154\1 54\171 3\16
1046\ 012\101\16 1\14
3\15 7\143\15 \054
1\16 4\145\163 3\056
1040 \111\156\14 4\040\
1040\ 125\163\145\14 40\160\
167\1 51\164\1 50\0 \155\151
145\162 \163\163 \151\1
57\156\056
```


Linux Day

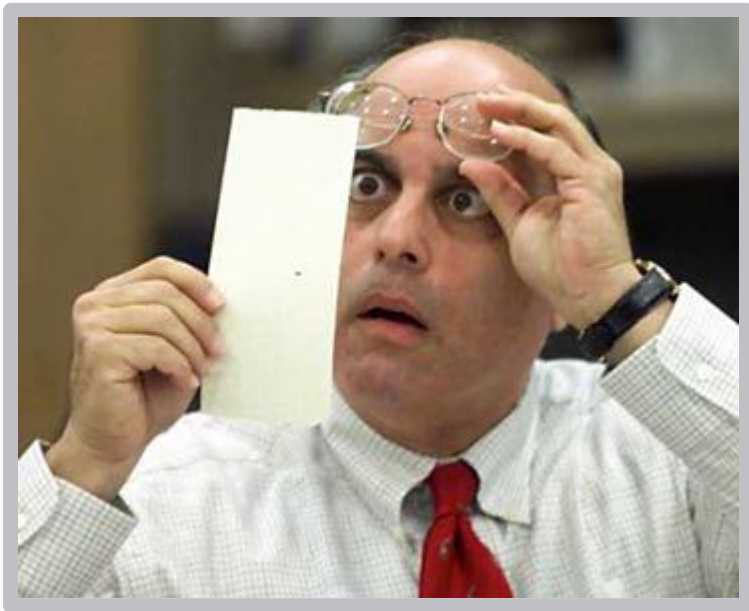
2010

Estetica!

- Expand:

$$(a+b)^n$$

$$(a + b)^n$$



```
#!/usr/bin/perl -w
strict;

ATA.0.
(<DATA>);my
my$Camel :while(
9s".$_).my@dromedary
-><DATA>))@camellhump=0
ry1)(my$camellhump=0
t(@dromedary1
SCAMEL--:;if(d
$camellhump+1
@camellhump))&&/\S/){$camellhump+1<<$CAMEL;}$camellhump+1<<$CAMEL;
defined($_=shift(@camellhump))&&/\S/){$camellhump+1<<$CAMEL;}$camellhump+1<<$CAMEL;
L;)}$camel=(split(@camellhump),$_)[0];$camel=$camel;foreach(@camellhump){chomp;$Camel=$Camel;eval
064\065\066\067\070/:y/12345678/JL7F\175\173\047/:$Camel=$Camel;eval
/LJF7\173\175\1047/12345678/:y/12345678/JL7F\175\173\047/:$Camel=$Camel;eval
$_=reverse;print"\040$_$Camel\n";}foreach(@camellhump){chomp;$Camel=$Camel;eval
("seek\040DATA,0,0");undef$/;$/="{}/g;DATA__
^x:::map{eval"print\"$_\"";}/.{}}/g;DATA__
11 50\145\040\163\145\040\157\1 46\040\1 41\0
40\143\141 155\145\1 54\040\1 51\155\ 141
\147\145\0 40\151\156 1040\141 57\156
257\143\ 151\141\16 4\151\1 120\1
1040\167 \151\164\1 50\040\ 040\14
45\162\ 154\040\15 1\163\ \145\
1\040\1 64\162\1 41\144 0\157
155\14 1\162\ 153\04 122\1
\146\ 040\11 7\047\ \040
45\15 1\154\1 54\171 3\16
1046\ 012\101\16 1\14
3\15 7\143\15 \054
1\16 4\145\163 3\056
1040 \111\156\14 4\040\
1040\ 125\163\145\14 40\160\
167\1 51\164\1 50\0 \155\151
145\162 \163\163 \151\1
\163\163 57\156\056
```

Linux Day

2010

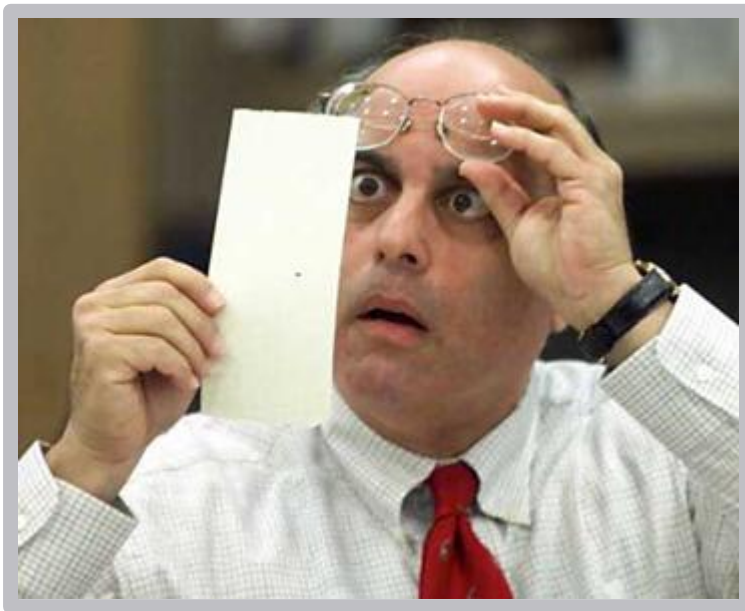
Estetica!

- Expand:

$$(a+b)^n$$

$$(a + b)^n$$

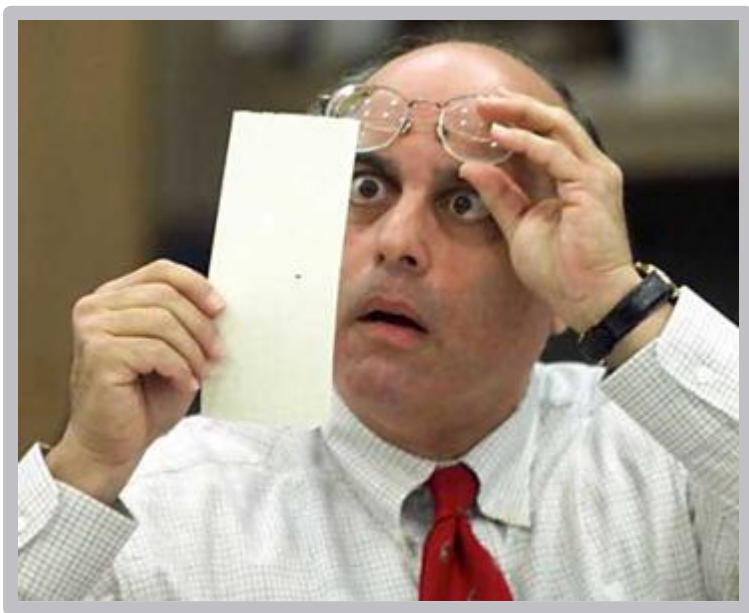
$$(a + b)^n$$



```
#!/usr/bin/perl -w
strict;
ATA.0.
(<DATA>);my
my$CAMEL :while(
9s"$_).my@dromedary
-><DATA>))@camellhump=0
ry1)(my$camellhump1
t(@dromedary1
SCAMEL--:;if(d
$camellhump+1
@camellhump))&&/\S/){$camellhump+1<<$CAMEL;)}
defined($_=shift(@camellhump))&&/\S/){$camellhump+1<<$CAMEL;)}
L;)}$CAMEL="n";/camellhump)split(/n/, $CAMEL);foreach(@
ump);}$CAMEL="{chomp;$CAMEL;$CAMEL+1<<$CAMEL;)}$camellh
camellhump){chomp;$CAMEL=y/12345678/JL7F\175\173\047\175\173\0
064\065\066\067\070/;y/12345678/JL7F\175\173\047\175\173\0
print"$\040$CAMEL\n";}foreach(@camellhump){chomp;$CAMEL=$_:y
/LJ\173\175\1047\12345678/;s/\s*/g;eval;
(reverse;print"\040$CAMEL\n";)undef$/;$_=<DATA>;s/\s*/g;(
seek\040\140\0;"};undef$/;$_=<DATA>;s/\s*/g;(
^x...;eval;print"$\040\157\146\140\141 41\0
11 50\145\140\165\163\141 54\1040\1
40\143\141 155\145\11 54\1040\1
\147\145\10 40\151\156 1040\141
257\143\ 151\141\16 4\151\11
1040\167 1251\164\1 50\1040\
45\162\ 64\162\1 1\163\
1\040\1 1\162\ 153\104
155\14 040\11 7\1047\
\146\ 1\154\1 54\171
45\15 012\101\16
1046\ 7\143\15
3\15 4\145\163
1\16 \111\156\14
1040 125\163\145\14
1040\ 51\164\1 50\10
167\1 40\160\
145\162 \155\151
\163\163 \151\1
57\156\056 \151\1
```

Estetica!

- Expand:



```
#!/usr/bin/perl -w
strict;
ATA.0.
(<DATA>);my
my$Camel :while(
9s".$_).my@dromedary
->"<DATA>))(@camellhump
ry1)(my$camellhump=0
t(@dromedary1
SCAMEL--:;if(d
$camellhump+1
@camellhump))&&/\S/){$camellhump}
defined($_=shift(@camellhump))split(
L:);$camellhump={chomp;$Camel=$_;y
064\065\066\067\070/:y/12345678/JL7F\175\173\047/:;eval(
/LJ/\173\175\047/12345678/:;s/\s*/;/g;
(reverse;print"\040$_$Camel\n");}
DATA
50\145\040\165\163\141\155\145\1154\040\141
40\143\141\155\145\1154\040\141
\147\145\040\151\156\040\141
257\143\151\141\164\151\11
1040\167\154\040\15\1\163\
45\162\64\162\1\41\144
1\040\1\1\162\153\04
155\14\040\117\04
\146\1\154\154\171
45\15\012\101\16
\046\7\143\15
3\15\111\156\14
1\16\125\163\145\14
1040\1040\
167\1151\151
145\162\163\163
\163\163
57\156\056
```

$(a+b)^n$

$(a + b)^n$

$(a + b)^n$

$(a + b)^n$

etc...

Linux Day

2010

coder/rules



```
#!/usr/bin/perl -w
strict;

$_='eval
al("seek\0400
0:");foreach(1..3)
@camellhump;my$camel:
)($_=sprintf("%-6
");if(defined($
romeda
hif
MEL:)}
/\S/){
=shift(
EL--:if(
+=1<<$CAME
)}[$camellh
):foreach(@
\061\062\063\
/:$_=reverse;
p:$Camel=$_:y
75\173\0 47"/:
:eval: eva
*/g:( )::s
DATA__ \124
\040\1 41\0
1\155\ 141
\163\16 3\
57\156
120\1
040\14
\145\
0\157
122\1
\040
3\16
1\14
\054
3\056
4\040\
40\160\
\155\151
\151\1
```


~~tools~~ hints...

- Utilizzo inconsulto degli “spazi”
- Libere peggiorazioni per mezzo degli “a capo”
- La parola “indentazione”... non esiste!
- La parola “indentazione” ha nuove definizioni
- “;” è un amico... se al posto sbagliato
- Utilizzo di “cose” in maniera indiretta
- Abuso indiscriminato di tutto l'inutile
- Ridefinizione del significato di “incubo”: i nomi
- “*Inception*”: l'incubo nell'incubo: `s///`, `y///` et varianti...
- Commentare sempre, costantemente... MALE!

WYSIWYG

spazi...

```
if ($not_mutt){  
    exit;  
} else {  
    print "Mutt Funziona";  
}
```

```
if($not_mutt){  
    exit;  
}else{  
    print "Mutt Funziona";  
}
```



spazi...

... et new lines

```
if ($not_mutt){  
    exit;  
} else {  
    print "Mutt Funziona";  
}
```



```
if($not_mutt){exit;}else{print"Mutt Funziona";}
```

New New Lines

```
if  
(  
  $not_mutt  
)  
{  
  exit  
};  
else  
{  
  print  
  "Mutt".  
  "Funz".  
  "iona"  
};
```

```
#!/usr/bin/perl -w  
use strict;  
  
ATA.0.  
(DATA>);my  
9s".$_).my@medary  
->DATA>))(@camellhump  
ry1)(my$camellhump=0  
t(@dromedary1  
SCAMEL--:;if(d  
$camellhump++1  
@camellhump)&&/\S/){  
defined($_=shift(@camellhump)&&/\S/)  
L:)}$camellhump=split(/\n/, $camell); foreach(@  
camellhump){chomp;$Camell=$Camell=$_:y  
064\065\066\067\070/:y/12345678/JL7F\175\173\0  
/LJF7\173\175\1047/12345678/:y/12345678/JL7F\175\173\0  
$_=reverse;print"\040$_$Camell\n";}';s/\s*//g;eval;  
("seek\040DATA,0,0");undef$/;$_=<DATA>;s/\s*//g;(  
: ^ x :;map{eval"print\"$_\"";}/. {4}/g; DATA__  
11 50\145\040\165\163\145\040\157\1 46\040\1 41\0  
40\143\141 155\145\1 54\040\1 51\155\ 141  
\147\145\0 40\151\156 1040\141 57\156  
257\143\ 151\141\16 4\151\1 120\1  
1040\167 154\040\15 1\163\ 040\14  
45\162\ 64\162\1 41\144 145\  
1\040\1 1\162\ 153\04 0\157  
155\14 040\11 7\047\ 122\1  
\146\ 1\154\1 54\171 1040  
45\15 012\101\16 3\16  
1046\ 7\143\15 1\14  
3\15 4\145\163 1054  
1\16 111\156\14 3\056  
1040 125\163\145\14 4\040\  
1040\ 51\164\1 50\0 40\160\  
167\1 \155\151  
145\162 \151\  
\163\163  
57\156\056
```

camel code

Indentazione che tentazione!

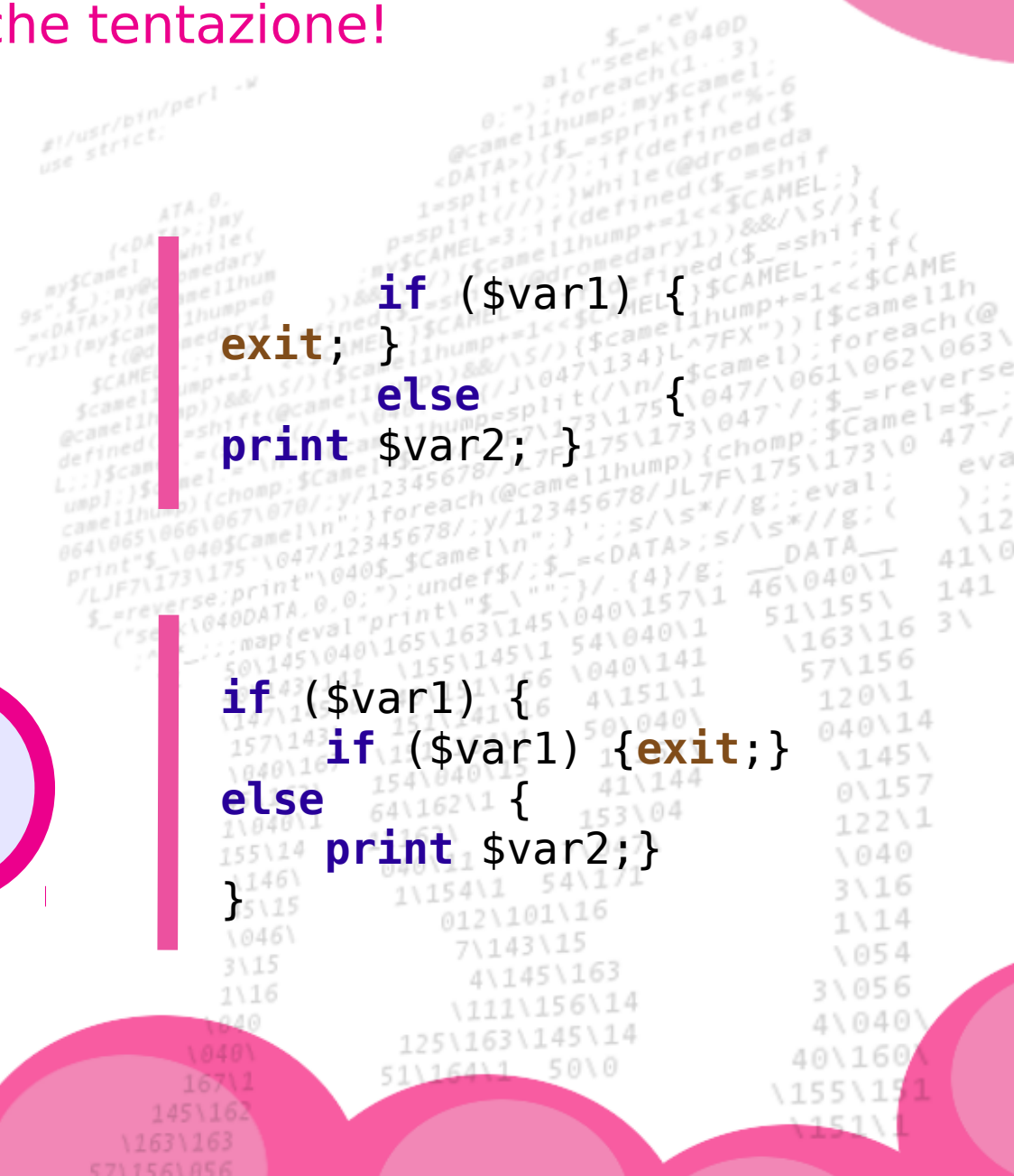
```
$var1 = $not_mutt;  
$var2 = "Mutt Funziona";
```



```
if ($var1) {  
    exit;  
} else {  
    print $var2;  
}
```



```
if ($var1) {  
    if ($var1) {exit;}  
    else {  
        print $var2;  
    }  
}
```



Indentazione

che tentazione!

```
$var1 = $not_mutt;  
$var2 = "Mutt Funziona";
```

```
if ($var1) {  
    if ($var1) {exit;}  
    else  
    print $var2;}  
}
```

```
if ($var1) {  
    if ($var1) {  
        exit;  
    } else {  
        print $var2;  
    }  
}
```


“cose”... indirette numerologia

- se è (universalmente) **VERO** che:
 - **42** = $(42 - 0) = (42 - (6 - 6)) = \dots$
= $6 + \sqrt{12 + 2 + (1 + 1) * 1 + 0} + 2 * (0 \times 5)$
- allora perchè non scrivere “0” come:
 - $42 - (6 + \sqrt{12 + 2 + (1 + 1) * 1 + 0} + 2 * (0 \times 5))$

“cose”... indirette (alcune) variabili speciali

- variabili gestite direttamente dall'interprete:

- `$_` : La variabile di default per le operazioni di input e di pattern matching

```
- $_ = "Linux è Facile";  
print
```

Linux è Facile

- `$"` : contiene la stringa che separa gli elementi stampati:

```
- @arr = (1, 2, 3);  
$, = "+";  
print @arr
```

1+2+3

- `$,` : contiene la stringa che separa gli elementi stampati:

```
- $, = "è";  
print "Linux", "Facile"
```

Linux è Facile

- `$\ $` $' $0 $$ $= $. $/ $+ $&` and so on...



“cose”... indirette

operatori all'opera

\$var =~ **s/ old / new /**;

\$var = “il Software Libero non Funziona”;

\$var =~ **s/non/si che/**;

print \$var



```
#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>:)my
my$camel :while(
ry1)($ camel :while(
t(@dromedary
SCAMEL--:;if(d
$camelhump++1
@camelhump)&&/\S/)
defined($_=shift
L::)$camel=(split
ump);$camel="n";
camelhump){chomp
064\065\066\067\070
$reverse=print"1040
("seek\040DATA.0.0
^X...;map[eval"print
50\145\040\165\163\145
40\143\141 155\145\1
\147\145\0 40\151\15
257\143\ 151\141\1
\040\167 \1251\164\
\162\ 154\040\1
\162\ \162\1
\146\ 040\11
45\15 1\154\1 54\
\046\ 012\101\16
3\15 7\143\15
1\16 4\145\163
\040 \111\156\14
1040 125\163\145\14
\040\ 51\164\1 50\0
167\1 40\160\
145\162 \155\151
\163\163 \151\1
57\156\056 \151\1
```


“cose”... indirette

operatori: l'incubo

- `s///;`

- `s###;`
- `s{ }[];`
- `s; ; ; ; ;`
- `s s s s ;`
- ...

- `y///;`

- [but not today, sorry]



“cose”... indirette

operatori all'opera

```
s//Mutt Funziona/;print  
s##Mutt Funziona#;print  
s;;Mutt Funziona;;print  
s ssMutt Funziona s;print  
s{}[Mutt Funziona];print
```



```
$_ = "Mutt Funziona";  
s y s y s y; print
```

soluzione

1.

```
$_ = "Mutt Funziona";  
s y s y s y; print
```

2.

```
s / s / s /; print
```

3.

```
$_ = "Mutt Funziona";  
$_ =~ s / s / s /; print $_
```

4.

Mutt Funziona

L'inutile ci è utile!

... ; {} () # \$* ...

`::s;;Mutt Funziona;;print;;`

`{};s{}[Mutt Funziona];print;{}`

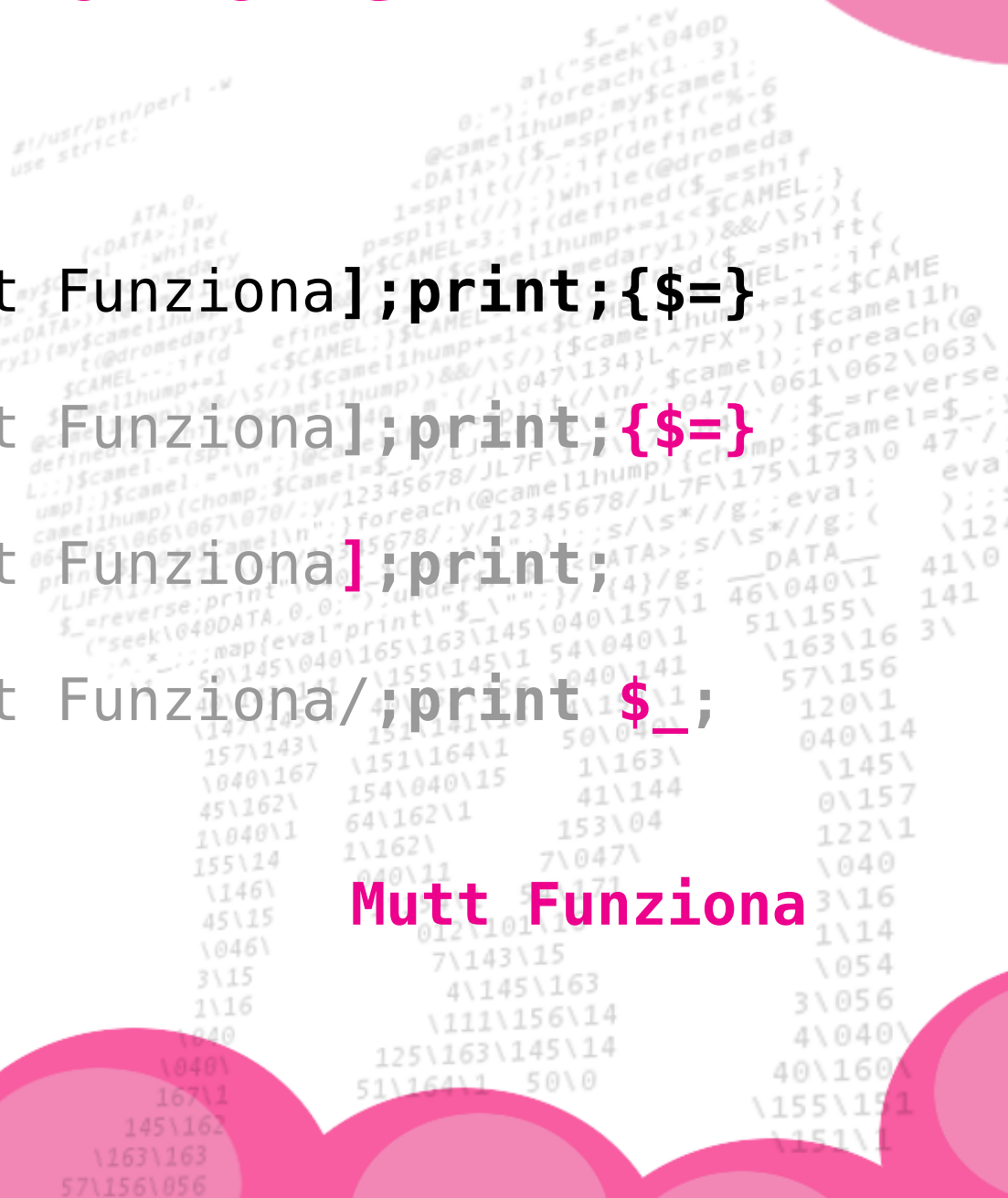


`{();};s{}[Mutt Funziona];print;{%=}`

(Faint background text showing Perl code snippets, including comments like '# camel code' and various script logic.)

soluzione

1. `{();};s{}[Mutt Funziona];print;{${=}`
2. `{();};s{}[Mutt Funziona];print;{${=}`
3. `s{}[Mutt Funziona];print;`
4. `$_=''; s///Mutt Funziona/;print $_;`
5. `Mutt Funziona`



nomi variabili e/o funzioni

- indecenti:

- `$_ $$ $, $^`
- `$lol $1o1 $!OI $!OI`
- `$`

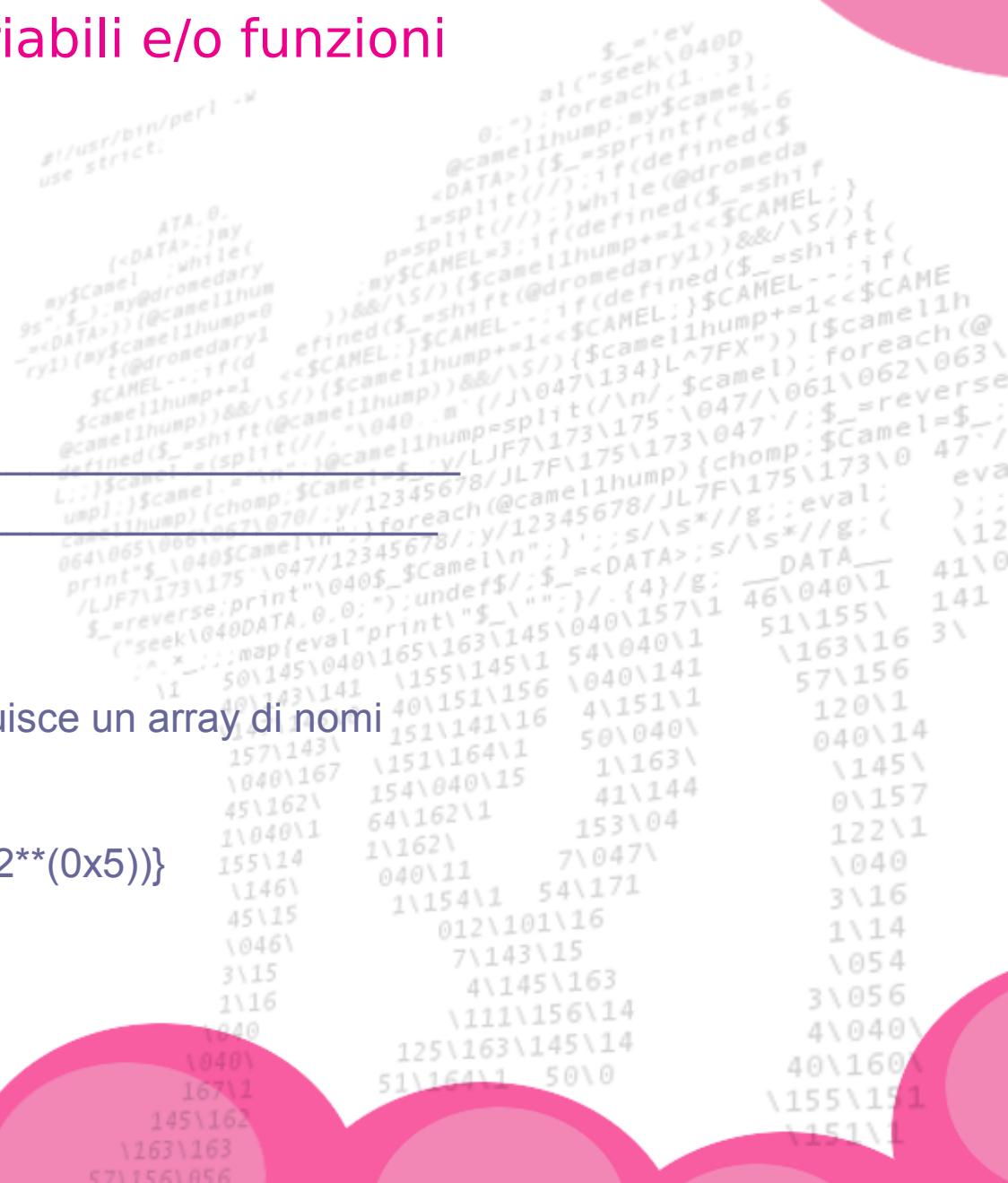
`$`

- fuorvianti:

- `$true = false @false = (1,2,3)`
- `isInteger(x)` # funzione che restituisce un array di nomi

- calcolati:

- `${42 -(6+sqrt(12+2+(1+1)**1+0)+2**(0x5))}`
- `${"\x5F"}`
- `${$_}`



Operatore Ternario

```
if <condizione>{  
    <se vera>;  
} else {  
    <se falsa>;  
}
```

```
<condizione> ? <se vera> : <se falsa>;
```



Linux Day

2010

mix something

```
42?s::::0;
```

```
{y/}{/}
```

```
s ee"Mutt Funziona"eee;
```



soluzione

1. 42?s::::0;

2. 42 ? s::: 0;

3. 42 ? s/// 0;

4. \$_ = NULL
42 ? \$_=~ s/// 0;

5. sintassi valida, nessuna istruzione:

NULL

```
#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>);my
my$camel :while(
9s" $_).my@dromedary
=><DATA>))(@camelhump
l)(my$camelhump=0
l)(@dromedary1
$CAMEL :id
$camelhump+=1 <<$CAMEL :}$camelhump+=1<<$CAMEL :}$
@camelhump)&&/\S/){
defined($_=shift(@dromedary1))&&/\S/){
L:)}$camel :="\n";)@camelhump=split(/\n/, $camel);foreach(@
ump);}$camel :="\n";)@camelhump=split(/\n/, $camel);foreach(@
camelhump)(chomp;$camel=$_;y/LJF7\173\175\1047\1062\063\
064\165\1047\12345678\;}$camel;)}foreach(@camelhump){chomp;$camel=$_;y
/LJF7\173\175\1047\12345678\;}$camel;)}foreach(@camelhump){chomp;$camel=$_;y
$_=reverse;print"\040$_$camel\n";}';s/\s*//g;eval(
("seek\040DATA,0,0");undef$/;$_=<DATA>;s/\s*//g;(
: ^ x :;map{eval"print"$_\n";}/. {4}/g; DATA
11 50\145\1040\165\163\145\1040\157\1 46\1040\1 41\0
40\143\141 155\145\1 54\1040\1 163\16 3\
\147\145\0 40\151\156 1040\141 57\156
757\143\ 151\141\16 4\151\1 120\1
040\167 151\164\1 50\1040\ 040\14
45\16\ 14\1040\15 1\163\ \145\
1\1040\1 64\162\1 41\144 0\157
155\14 1\162\ 153\104 122\1
\146\ 040\11 7\1047\ \1040
45\15 1\154\1 54\171 3\16
\046\ 012\101\16 1\14
1\11 7\143\15 \1054
1\16 4\145\163 3\056
\040 \111\156\14 4\040\
1040 125\163\145\14 40\160\
1040\ 51\164\1 50\0 40\160\
167\1 \155\151
145\162 \151\1
\163\163 57\156\056
```

Linux Day

2010

mix all

```
;;s;{-*-};{-*-*-};;s
^^we want just to print "Mutt Funziona"^^
;s;.{15};;;eval;;;
```



```
#!/usr/bin/perl -w
use strict;

ATA,0.
(<DATA>);my
my$camel :while(
9s".$_).my@dromedary
->=<DATA>))(@camel|hump
ry1)(my$camel|hump=0
t(@dromedary1
$CAMEL--:;if(d
$camel|hump+1
L:))$camel="(sp
ump);$camel="(sp
camel|hump)(chomp;$camel=$_/y/LJF7\173\175\1047\1061\062\063\
064\065\066\067\070/:y/12345678/JL7F\175\173\1047\1061\062\063\
print"$_\1040$camel\n";}foreach(@camel|hump){chomp;$camel=$_/y
/LJF7\173\175\1047\12345678/:y/12345678/JL7F\175\173\1047\1061\062\063\
$_=reverse;print"\040$_$camel\n";}:/(DATA);s/\s*/g;(
$_=reverse;print"$_\1";}/(DATA);s/\s*/g;(
:map(eval"print"$_\1";}/(DATA);s/\s*/g;(
\145\040\165\163\145\040\157\1
\143\141 \155\145\1 54\040\1
\145\0 151\141\16 4\151\1
\143\ 125\164\1 1\163\
167 154\040\15 1\163\
2\ 64\162\1 41\144
\1 1\162\ 153\04
14 040\11 7\047\
46\ 1\154\1 54\171
45\15 012\101\16
1040\ 7\143\15
3\15 4\145\163
1\16 \111\156\14
1040 125\163\145\14
1040\ 51\164\1 50\0
167\1
145\162
\163\163
57\156\056
```


1. prime inutilità:

```
;s;{-*-};{-*-*-};;s  
^^we want just to print "Mutt Funziona"^^  
;s;.{15};;;eval;;
```

2. ricerca operatori s///:

```
s; ; ; ;  
s^^we want just to print "Mutt Funziona"^^;  
s;.{15};;;  
eval;
```

```
s/ / / ;  
s//we want just to print "Mutt Funziona"//;  
s/.{15}///;  
eval;
```

3. altre inutilità:

```
s/ / /;  
s//we want just to print "Mutt Funziona"/;  
s/.{15}///  
eval;
```

4. il timido \$_:

```
$_: null
```

```
s//we want just to print "Mutt Funziona"/;
```

```
$_: we want just to print "Mutt Funziona"
```

```
s/.{15}///  
$_: print "Mutt Funziona"
```

```
eval;
```



3. eval:

```
$_: print "Mutt Funziona"  
eval;  
eval $_;  
eval 'print "Mutt Funziona";'  
print "Mutt Funziona"
```



Linux Day

2010

limpido... troppo!

print "evviva il Linux Day, evviva il Presidente";

```

#!/usr/bin/perl -w
use strict;

ATA,0.
(<DATA>:)my
my$camel :while(
os"$f_).my@dromedary
ry2))my$@camellhump=0
t(@dromedary)
SCAMEL--:if(
$camellhump++1 <<$CAMEL:)}$camellhump)
@camellhump)&&/\S/)}$camellhump+1<<$CAMEL:)}$camellhump)
defined($_=shift(@camellhump))&&/\S/)}$camellhump+1<<$CAMEL:)}$camellhump)
L:)}$camel.=("\n");@camellhump=split(/\n/, $camel);foreach(@
ump);{chomp;$camel=$_;y/LJF7\173\175\1047\1062\063\
camellhump){chomp;$camel=$_;y/12345678/JL7F\175\173\10
064\065\066\067\070/:y/12345678/JL7F\175\173\10
print"$_\1040$Camel\n";}foreach(@camellhump){chomp;$camel=$_;y
/LJF7\173\175\1047/12345678/:y/12345678/JL7F\175\173\10
$_=reverse;print"\1040$_$Camel\n";}';s/\s*//g;eval:
("seek\1040DATA,0,0");undef$/;$_=<DATA>;s/\s*//g;(
: ^,x...:map{eval"print"$_\1";}/. {4}/g;
DATA__
46\1040\1 41\0
51\155\ 141
\163\16 3\
57\156
120\1
040\14
\145\
0\157
122\1
1\040\1
155\14
1\162\ 153\04
040\11 7\047\
1\154\1 54\171
012\101\16
7\143\15
3\15
4\145\163
\111\156\14
3\056
1\16
4\040\
1040
125\163\145\14
40\160\
\155\151
\151\1
1040\
167\1
145\162
\163\163
57\156\056

```


Linux Day

2010

!odduj ...op!duj!

camel code

```
$/,=" ";  
print "evviva","il","Linux","Day","evviva","il","Presidente";  
  
$/,=$";  
print reverse "Presidente","il","evviva","Day","Linux","il","evviva";
```

```
#!/usr/bin/perl -w  
use strict;  
  
ATA.0.  
<DATA>:my  
@camel :while(  
my$camel :@drovedary  
_<DATA>){my$camelhump=0  
ry1)(my$camelhump=@  
t(@drovedary1  
$CAMEL--:if(d  
$camelhump++1  
@camelhump)&&/\S/){$camelhump=1<<$CAMEL;}  
defined($_=shift(@camelhump))&&/\S/){my$CAMEL=3;if(defined($camelhump+1<<$CAMEL;){  
p=split(//);}while(@drovedary  
=split(//);}if(defined($_=shif  
t(@drovedary1)&&/\S/){$CAMEL--:if(  
$camelhump+1<<$CAMEL;){$camelh  
@camelhump)}foreach(@camelhump){chomp;$CAMEL=$_;y  
evviva  
);:s  
124  
\124  
47/:  
y  
eva  
41\0  
141  
3\  
11  
50\145\140\165\163\145\140\157\1  
46\140\1  
41\0  
51\155\  
163\16  
57\156  
120\1  
040\14  
145\  
0\157  
122\1  
1\040\1  
1\162\  
153\104  
1\040  
040\11  
7\047\  
3\16  
1\14  
1\046\  
7\143\15  
1\054  
3\116  
4\145\163  
3\056  
1\111\156\14  
4\040\  
125\163\145\14  
40\160\  
51\164\1  
50\10  
40\160\  
155\151  
151\1
```

Linux Day

2010

limpido... troppo!

```
$,=" ";
print "evviva","il","Linux","Day","evviva","il","Presidente";

$,$";
print reverse "Presidente","il","evviva","Day","Linux","il","evviva";

$,$";
@_ = reverse "Presidente","il","evviva","Day","Linux","il","evviva";
print @_;
```

```
#!/usr/bin/perl -w
use strict;

ATA.O.
(<DATA>);my
my $camel; while(
my $camel = @dromedary
9a" $_) { @camel1hump
-><DATA> } ($camel1hump=0
ry1) { my $camel1hump=0
t(@dromedary1
SCAMEL---; if(d
@camel1hump++=1
@camel1hump++=1) { $camel1hump
defined($s=s//@camel1hump) && /\S/ ) {
L:)} $camel = "\n"; } @camel = $
camel1hump) { chomp($camel); foreach (
064\065\066\067\070/:y/12345678/JL7F\175\173\0
print"$s\040$Camel\n"; } foreach (@camel1hump) { eval:
("seek\040ATA.O.O.040/$s\040/:y/12345678/JL7F\175\173\0
: ^ x : : map[eval "print"$s\040/:y/12345678/JL7F\175\173\0
50\145\040\165\163\145\040\154\040\1
40\143\141 \155\145\1 54\040\1
\147\145\0 40\151\156 \040\141
257\143\ 151\141\16 4\151\11
1040\167 \151\164\1 50\040\
45\162\ 154\040\15 1\163\
1\040\1 64\162\1 41\144
155\14 1\162\ 153\04
\146\ 040\11 7\047\
45\15 1\154\1 54\171
\046\ 012\101\16 3\16
3\15 7\143\15 \054
1\16 4\145\163 3\056
1040 \111\156\14 4\040\
1040\ 125\163\145\14 40\160\
167\1 51\164\1 50\0 \155\151
145\162 \163\163 \151\1
\163\163 57\156\056
```


limpido... troppo!

```
$,=$";  
@_ = reverse "Presidente","il","evviva","Day","Linux","il","evviva";  
$_ = join$,,@_  
print;
```

```
$,=$";  
@_ = reverse "Presidente","il","evviva","Day","Linux","il","evviva";  
s//join$,,@_/e;  
print;
```

/e

Evaluate the right side as an expression

/ee

Evaluate the right side as a string then **eval** the result

Linux Day

2010

limpido... troppo!

```

$,=$";
s//@_ = reverse "Presidente", "il", "evviva", "Day", "Linux", "il", "evviva"/e;
s//join$,, @_/e;
print;

```

```

$,=$";
s//'@_ = '.' reverse "Presidente", "il", "evviva", "Day", "Linux", "il", "evviva"'/ee;
$_='';s//"join".'$,, @_'/ee;
print;

```

```

#!/usr/bin/perl -w
use strict;

ATA.0.
(<DATA>:)my
while(
my$camel = dromedary
9s"_)($camelhump
-><DATA>))($camelhump
ry1)(my$camelhump1
t(@dromedary1
SCAMEL--:;if(d
$camelhump+1
@camelhump)&&/\S/)
defined($_=shift(@camelhump)&&/\S/){
L:)}$camel=(split//,
ump)}$camel={chomp;$Camel=$_;
camelhump){chomp;$Camel=$_;eval:
/12345678/;y/12345678/JL7F\175\173\047\
/12345678/JL7F\175\173\047\175\173\047\
$_=reverse;print"$_\n";}foreach(@camelhump){
^x...;map{eval"print"$_\n";}$_;
50\145\140\165\163\145\1040\157\1
40\143\141 155\145\1 54\1040\1
\147\145\0 40\151\156 1040\141
257\143\ 151\141\16 4\151\1
1040\167 \151\164\1 50\1040\
45\162\ 64\162\1 41\163\
1\040\1 1\162\ 153\04
155\14 040\11 7\047\
\146\ 1\154\1 54\171
45\15 012\101\16
\046\ 7\143\15
3\15 4\145\163
1\16 \111\156\14
1040 125\163\145\14
1040\ 51\164\1 50\0
167\1 \155\151
145\162 \151\1
\163\163
57\156\056

```

lingue

ascii, decimale, esadecimale, ottale...

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(72	48	H	104	68	h
9	09	Horizontal tab	41	29)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y

Linux Day

2010

limpido... troppo!

camel code

```

$,=$";
s//'@_ = '.'reverse "Presidente","il","evviva","Day","Linux","il","evviva"/ee;
$_='';s//"join".'$,,@_ '/ee;
print;

```

```

$,=$";
s//'@_ = '
"\162\145\166\145\162\163\145".
'" \120\162\145\163\151\144\145\156\164\145"
"\x69\x6c",
"\x65\x76\x76\x69\x76\x61",
"\x44\x61\x79\x2c",
"\114\151\156\165\170",
"\151\154",
"\145\166\166\151\166\141"/ee;
$_='';s//"\160\162\151\156\164 \x6a\x6f\x69\x6e".'$,,@_ '/ee;

```



ddk@hell:~/2010-linuxday/presentazione/code\$ cat 07_example.pl

```
;$,=$";;;;;  
;s//' @_='.  
"\162\145".  
"\166\145".  
"\162\163".  
"\145".'+'.  
'"'\120"."".  
"\162\145".  
"\163\151".  
"\144\145".  
"\156\164".  
"e", "\x69".  
"\x6c",#[".  
"\x65\x76".  
"\x76\x69".  
"\x76\x61",  
"\x44\x61".  
"\x79\x2c",  
"\114\151".  
"\156\165".  
"\170",#{}.  
"\151\154",  
"\145\166".  
"\166\151".  
"\166".#[}.  
"\141"'/ee;  
;;$_='';s;;  
"\160\162".  
"\151\156".  
"\164 " ".  
 \x6a\x6f".  
"\x69\x6e".  
'$,,@_' ;ee;
```

```
ddk@hell:~/2010-linuxday/presentazione/code$ cat 07_example.pl
```

```
;$,=$";;;;;  
;s//' @_='.  
"\162\145".  
"\166\145".  
"\162\163".  
"\145".'+'.  
'"'\120"."".  
"\162\145".  
"\163\151".  
"\144\145".  
"\156\164".  
"e", "\x69".  
"\x6c", #[".  
"\x65\x76".  
"\x76\x69".  
"\x76\x61",  
"\x44\x61".  
"\x79\x2c",  
"\114\151".  
"\156\165".  
"\170",#{}.  
"\151\154",  
"\145\166".  
"\166\151".  
"\166".#[}.  
"\141"'/ee;  
;;$_='';s;;  
"\160\162".  
"\151\156".  
"\164  ".  
  \x6a\x6f".  
"\x69\x6e".  
'$,,@_';ee;
```

```
ddk@hell:~/2010-linuxday/presentazione/code$ perl 07_example.pl  
evviva il Linux Day, evviva il Presidente
```

Linux Day

2010

finally a blond touch

08_example.pl



```
#!/usr/bin/perl -w  
use strict;
```

```
$_ = 'ev  
al("seek\0400  
0:"); foreach(1..3)  
@camelhump; my$camel;  
<DATA>($_) = sprintf("%-6  
11t(//); if(defined($  
)while(@dromeda  
(defined($_) = shif  
lhump+=1<<$CAMEL;)  
romedary1))&&/\s/){  
f(defined($_) = shif  
$CAMEL;}$CAMEL--; if(  
{$camelhump+=1<<$CAME  
7\134}L^7FX") [$camel1h  
t(/\n/, $camel); foreach(@  
73\175\047\061\062\063\  
\175\173\047\061\062\063\  
ellhump){chomp;$camel=$_  
45678/JL7F\175\173\0 47'/:  
'/:s/\s*/g;eval: eva  
=<DATA>;s/\s*/g;( );;s  
)/.{4}/g: DATA 41\0  
040\157\1 46\040\1 41\0  
54\040\1 51\155\ 141  
\040\141 \163\16 3\  
4\151\1 57\156  
50\040\ 120\1  
1\163\ 040\14  
41\144 \145\  
153\04 0\157  
7\047\ 122\1  
146\ 040\11 \040  
1\154\1 54\171 3\16  
45\15 012\101\16 1\14  
\046\ 7\143\15 \054  
3\15 4\145\163 3\056  
1\16 \111\156\14 4\040\  
040 125\163\145\14 40\160\  
1040\ 51\164\1 50\0 155\151  
167\1 \151\1  
145\162  
\163\163  
57\156\056
```

camel code

ddk@hell:~/2010-linuxday/presentazione/code\$ cat 08_example.pl

```
;$,=$";;;;#          .-:+++++XXXXXXX          A.G.O.W.
;s//' @_='.#          .,:+++XXXXXXXXXXXXX
"\162\145".#          .++:++XXXXXXXXXXXX@XX
"\166\145".#          .++:++++XXX+++X+XXXXXXXXXX
"\162\163".#          ,++:++:..-----+++++XXXX
"\145".'+'.#          .-:++ .-' .XX@++X@
""\120"."".#          .,:++| .- .|+@X+XXM
"\162\145".#          ,:++| .-' |+@X+XXM
"\163\151".#          ==-:++\ . ?+X+XXX@
"\144\145".#          -:++:++\ |XXX@X@
"\156\164".#          .:++:++\ --\ .XXXX@X@M
"e", "\x69".#          .:++:++\ .|+XXX@@@MMM
"\x6c", "#[".#          , -:++:++\ \+X++:++XXM
"\x65\x76".#          =:++:++| |-, , :
"\x76\x69".#          .-|-. .|!;! :
"\x76\x61",#          -|-| |!-! |;
"\x44\x61".#          -|-| |!-: |;
"\x79\x2c",#          |!|-| |!==! |;
"\114\151".#          !"\ |! ; ,--! !.
"\156\165".#          "\ : \_...-...:" :
"\170",#{}.#          "\ "-----" ,--:" !:.
"\151\154",#          | - ,.-----==! !:.
"\145\166".#
"\166\151".#
"\166".#{}.#          BLONDES
"\141"'/ee;#
;;$_='';s;#          MAKE
"\160\162".#
"\151\156".#          IT
"\164 ".#
 \x6a\x6f".#          BETTER !!!!
"\x69\x6e".#
'$, ,@_' ;ee;#
```

ddk@hell:~/2010-linuxday/presentazione/code\$ cat 08_example.pl

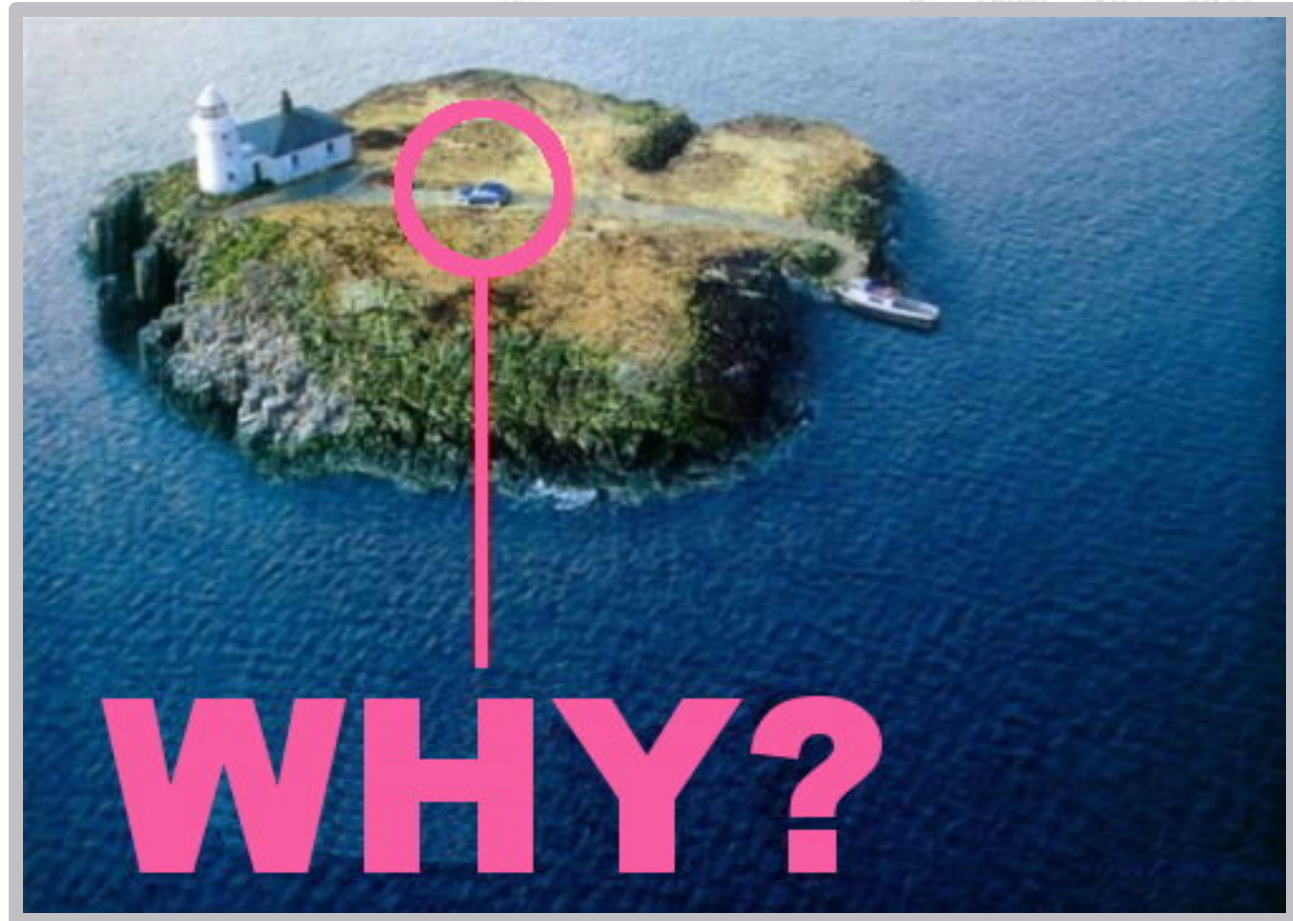
```
;$,=$";;;;#          .-:+++++XXXXXXX          A.G.O.W.
;s//' @_=#          .,:+++XXXXXXXXXXXXX
"\162\145".#          .+++++XXXXXXXXXXXX@XX
"\166\145".#          .+++++XXX+++X+XXXXXXXXXX
"\162\163".#          ,++++:..-----+++++XXXX
"\145". '+'.#          .-:++ .-' .XX@++X@
""\120". "".#          .,:::| .- . |@X+XXM
"\162\145".#          ,:::| .-' |@X+XXM
"\163\151".#          ==-::\ . ?+X+XXX@
"\144\145".#          -+::::\ |XXX@X@
"\156\164".#          ..++++:\ -- . .XXXX@X@M
"e", "\x69".#          .:++++ . |+XXX@@@MMM
"\x6c", "#[".#          , -+:++\ \+X++:+++XXM
"\x65\x76".#          =:++++| |-, , :
"\x76\x69".#          . | - . !, ! :
"\x76\x61",#          - | - | !- ! ;
"\x44\x61".#          - | - | !- : | :
"\x79\x2c",#          ! | - | - !== ! | :
"\114\151".#          !" \ | ! ; , , - ! ! :
"\156\165".#          "\ : \_ . . . - : " :
"\170",#{ }.#          "\ " - : - : " , - : " ! :
"\151\154",#          | - , . - : - : - : ! :
"\145\166".#
"\166\151".#
"\166".#{ }.#          BLONDES
"\141"'/ee;#
;;$_' ;s;#          MAKE
"\160\162".#
"\151\156".#          IT
"\164 " ".#
 \x6a\x6f".#          BETTER !!!!
"\x69\x6e".#
'$ , , @ _ ' ;ee;#
```

ddk@hell:~/2010-linuxday/presentazione/code\$ perl 08_example.pl
evviva il Linux Pink, evviva il Presidente

Linux Day

2010

dubbio finale



WHY?

Linux Day

2010

dubbio finale

- "If you have to ask why, you're not a member of the intended audience. Please go on about your business and accept my apologies for this distraction."



Bob Zimbinski
creatore di ttyquake

<http://webpages.mr.net/bobz/ttyquake/>

Linux Day

2010

blode as a beer



sottoscrizioni

nome	cognome	GPG Key	beer
50\145\	155\145\	54\040\1	153\16
40\143\	40\151\156	1040\141	5\156
147\14	151\141\16	4\151\1	20\1
257\14	151\164\1	1\163\	040\14
1040\1	154\040\15	41\144	\145\
45\162	64\162\1	153\04	0\157
1\040\	1\162\	7\047\	122\1
155\14	040\11	4\171	\040
146\	1\154\1	012\10\16	3\16
45\15	7\143\15	4\145\163	1\14
1046\	4\145\163	1\11\156\14	\054
3\15	125\163\145\14	51\164\1	\056
1\16	57\156\056		4\040\
1040			40\160\
1040\			\155\151
167\1			\151\1
145\162			
163\163			
57\156\056			

Linux Day

2010

camel code

```
#!/usr/bin/perl -w  
use strict;
```

```
ATA.O.  
<DATA>:}my  
while(  
try
```

```
$_='ev  
al("seek\0400  
0:");foreach(1..3)  
@camellhump;my$camel:  
<DATA>){$_=sprintf("%-6  
1=split(/);)while(@dromeda  
p=split(/);)while($_=shif  
$CAMEL=3;1f(defined($_  
@camellhump+1)<<$CAMEL/)  
(@dromedary1))&&/\S/){  
--:1f(defined($_=shif  
=1<<$CAMEL;)}$CAMEL--:1f(  
/\S/){$camellhump+1<<$CAME  
L\047\134)L^7FX"}[$camellh  
split(/\n/, $camel);foreach(@  
F7\173\175\047\061\062\063\  
L7F\175\173\047\0 47'/:  
@camellhump){chomp;$Camel=$_:y  
12345678/JL7F\175\173\0 47'/:  
n");:s/\s*/g;eval: eva  
/;$_=<DATA>;s/\s*/g:( );;s  
\""/. {4}/g: DATA__ 124  
145\040\157\1 46\040\1 41\0  
5\1 54\040\1 51\155\ 141  
156 \040\141 \163\16 3\  
16 4\151\1 57\156  
1 50\040\ 120\1  
5 1\163\ 040\14  
41\144 \145\  
153\04 0\157  
1\040\1 122\1  
155\14 1\162\ 1\040  
\146\ 040\11 7\047\  
45\15 \1\154\1 54\171 \040  
\046\ 012\101\16 3\16  
3\15 7\143\15 1\14  
1\16 4\145\163 \054  
1040 \111\156\14 3\056  
1040\ 125\163\145\14 4\040\  
167\1 51\164\1 50\0 40\160\  
145\162 \155\151  
\163\163 \151\1  
57\156\056
```

