

Diese HTML-Datei enthält spezielle CSS2-Anweisungen, die für den [Präsentationsmodus](#) des Web-Browsers [Opera](#) erstellt wurden. Sofern sie diesen Browser benutzen, können sie mit der F11- Taste in diesen Modus (und wieder zurück) schalten. Die Präsentation wurde für eine Auflösung von 1280x1024 Bildpunkten erstellt.



Weird programming

Programmieren, wie es die Welt nicht braucht

- Vortragsreihe "Chaos-Seminar"
- Veranstalter: CCC, Erfa-Kreis Ulm
 - <http://www.ulm.ccc.de/>
 - mail@ulm.ccc.de
 - Montagstreff: ab ca 19:30, Cafe Einstein (Uni)
 - Anfahrtsskizze auf der Webseite
- Referent: Markus Schaber
 - <http://www.schabi.de/>
 - markus.schaber@ulm.ccc.de
- Vortrag ist im HTML-Format online
 - <http://www.ulm.ccc.de/~schabi/weirdprog/>

Inhaltsübersicht

- Hello World
- 99 Bottles of Beer
- Quines
- Spaßige Programmiersprachen
- Obfuscated Programming
- Selbst modifizierender Code

Hello World

- Beliebte Aufgabe im Bereich "mein erstes Programm"
- Wohl am häufigsten gelöste Programmieraufgabe
- Sammlung des ACM
 - <http://www2.latech.edu/~acm/HelloWorld.shtml>
 - Über 200 Programmiersprachen
- Hier ein paar heute eher "exotische" Programmiersprachen

Hello-World-Exoten:

- Caml

```
let printHelloWorld() =  
  print_string "Hello World\n";;
```

- Haskell


```

C # data division.
C # procedure division.
C #
C # * ))cleartomark /Bookman-Demi findfont 36 scalefont setfont (
C # * (
C #
C # * hello polyglots$
C # main.
C # perform
C * ) 2>_$$; echo "hello polyglots"; rm _$$; exit
C print
C stop run.
C -*, 'hello polyglots'
C
C print.
C display "hello polyglots". (
C /* int i; /*
C /* main () { /*
C /* i=printf ("hello polyglots\n"); O= &i; return *O; /*
C *) (*
C *) begin (*
C *) writeLn ('hello polyglots'); (*
C *) (* )
C * ) pop 60 360 (
C * ) pop moveto (hello polyglots) show (
C * ) pop showpage ((
C *)
C end .(* )
C)pop% program polyglot. *){*/}

```

99 Bottles of Beer

- Fortgeschrittenenaufgabe in Programmierkursen
- Dient zur Einführung von Schleifen
- Erfunden im Usenet als Reaktion auf Troll
- <http://internet.is-la.net/mirrors/99bottles/> - derzeit 227 Beispiele
- Das historische erste Auftreten:

```

10 REM Basic version of 99 bottles of beer
20 FOR X=100 TO 1 STEP -1
30 PRINT X;"Bottle(s) of beer on the wall,";X;"bottle(s) of beer"
40 PRINT "Take one down and pass it around,"
50 PRINT X-1;"bottle(s) of beer on the wall"
60 NEXT

```

99 Bottles of Beer

- APL2-Version:

```

BEER;⍵IO;⍵PW
⍝APL2 version (no loops) of "99 bottles of beer" by Chuck Kennedy
⍵IO←0 ⍝Start counting from 0
⍵PW←150 ⍝Set printing width to 150 chars
a←,[10]⊖⊖100 ⍝Integers from 99 thru 0 as a column matrix
b←c' bottles of beer on the wall,'
c←c' bottles of beer,'
d←c' Take one down and pass it around.'
e←((a,b,a,c),d),(1⊖a),b ⍝Build the song, next two lines fix the English
e[99;1 3 4 5]←(c(⊃b)~'s')(c(⊃c)~'s')(c'Take it down and pass it around.')(c'No more')
e[99;]←'No more' b 'No more' c 'Go to the store and buy some more.' 99 b
e ⍝Display the song

```

99 Bottles of Beer

- Makefile

```
# quick effort at 99 bottles program using gnu make
#
# the file must be called makefile.bottles
#
# Author: Andrew Dunstan (andrew.dunstan@its.maynick.com.au)
#

default:
    $(MAKE) -f makefile.bottles BOTTLES=99 bottles

.SILENT:

bottles:
    echo $(BOTTLES) bottles of beer on the wall
    echo $(BOTTLES) of beer
    echo Take one down and pass it around
ifeq ($(BOTTLES),0)
    echo No bottles of beer on the wall
else
    echo `expr $(BOTTLES) - 1` bottles of beer on the wall
    echo
    $(MAKE) -f makefile.bottles BOTTLES=`expr $(BOTTLES) - 1` bottles
endif
```

Quines

- Ein Programm, das den eigenen Quelltext ausgibt
- Benannt nach Willard van Orman Quine, via Douglas Hofstadter
- Aufgabe ist eigentlich nicht so einfach:

```
print "hallo"

zu

print "print \"hallo\""

tut nicht!
```

- In BASIC trivial, da Zugriff auf den Quelltext

```
10 LIST
```

- Einfachstes Quine: Leeres Programm
 - zulässig z. B. in Bash, Python, Perl, HQ9+, K&R-C...
- Für ein "echtes" Quine ist aber etwas Trickserei notwendig

Einfache Quines

- Lisp, Scheme:

```
((lambda (x)
  (list x (list (quote quote) x)))
 (quote
  (lambda (x)
    (list x (list (quote quote) x)))))
```

- C

```
char*f="char*f=%c%s%c;main(){printf(f,34,f,34,10);}%c";main(){printf(f,34,f,34,10);}
```

- Perl

```
$_=q{$_=q{Q};s/Q/$_/;print};s/Q/$_/;print
```

- Python

```
_='_=%s;print _%%`_`;print _%`_`
```

Multilingual Quine

- Turbo Pascal und ANSI-C

```
(*a);main(){char i,*s[]={"%c%s%c%c%c",
"(*a);main(){char i,*s[]={",
"%c%c%c%s%c%s%c%s%c/%*%c%s",
";printf(s[1]);for(i=0;i<=12;i++)printf(s[0],34,s[i],34,44,10);",
"printf(s[2],34,34,125,s[3],10,s[4],10,s[5],10,s[6],10,41,s[7]);",
"for(i=0;i<=12;i++)printf(s[0],39,s[i],39,44,10);",
"printf(s[12],39,39,41,s[8],10,s[9],10,s[10],10,s[11],10,47,125);",
"const q=#34;w=#39;n=#13#10;s=array[0..13] of string=(",
";var i:integer;begin write(s[1]);for i:=0to 12do write(q,s[i],q,#44,n);",
"write(#34#34#125,s[3],n,s[4],n,s[5],n,s[6],n,#47#42#41,s[7]);",
"for i:=0to 12do write(w,s[i],w,#44,n); ",
"write(#39#39#41,s[8],n,s[9],n,s[10],n,s[11],n,#123#42#47#125);end.",
"%c%c%c%s%c%s%c%s%c/%*%c%c",
"});printf(s[1]);for(i=0;i<=12;i++)printf(s[0],34,s[i],34,44,10);
printf(s[2],34,34,125,s[3],10,s[4],10,s[5],10,s[6],10,41,s[7]);
for(i=0;i<=12;i++)printf(s[0],39,s[i],39,44,10);
printf(s[12],39,39,41,s[8],10,s[9],10,s[10],10,s[11],10,47,125);
/*)const q=#34;w=#39;n=#13#10;s=array[0..13] of string=('%c%s%c%c%c',
'(*a);main(){char i,*s[]={',
'%c%c%c%s%c%s%c%s%c/%*%c%s',
';printf(s[1]);for(i=0;i<=12;i++)printf(s[0],34,s[i],34,44,10);',
'printf(s[2],34,34,125,s[3],10,s[4],10,s[5],10,s[6],10,41,s[7]);',
'for(i=0;i<=12;i++)printf(s[0],39,s[i],39,44,10);',
'printf(s[12],39,39,41,s[8],10,s[9],10,s[10],10,s[11],10,47,125);',
'const q=#34;w=#39;n=#13#10;s=array[0..13] of string=',
';var i:integer;begin write(s[1]);for i:=0to 12do write(q,s[i],q,#44,n);',
'write(#34#34#125,s[3],n,s[4],n,s[5],n,s[6],n,#47#42#41,s[7]);',
'for i:=0to 12do write(w,s[i],w,#44,n); ',
'write(#39#39#41,s[8],n,s[9],n,s[10],n,s[11],n,#123#42#47#125);end.',
'%c%c%c%s%c%s%c%s%c/%*%c%c',
'');var i:integer;begin write(s[1]);for i:=0to 12do write(q,s[i],q,#44,n);
write(#34#34#125,s[3],n,s[4],n,s[5],n,s[6],n,#47#42#41,s[7]);
for i:=0to 12do write(w,s[i],w,#44,n);
write(#39#39#41,s[8],n,s[9],n,s[10],n,s[11],n,#123#42#47#125);end.
{*/}
```

Spassige Programmiersprachen

- Anscheinend nach dem Genuss von 99 Bottles of Beer entstanden
- Bereiten manchmal Kopfschmerzen
- Einige davon haben Obfuscation (siehe später) als Hauptziel

HQ9+

- Optimiert auf die wichtigsten Problemstellungen des Programmieranfängers:
 - H - gibt "Hello World" aus
 - Q - gibt den eigenen Quelltext aus
 - 9 - gibt 99 Bottles of Beer aus
 - + - erhöht das Arbeitsregister um 1
- Einige Beispiele:
 - Hello World
 - H
 - Quine 4-fach
 - QQQQ

Brainfuck

- Erste Implementierung von Urban Müller für den Amiga
- Brainfuck-Compiler für AmigaOS 2.0: nur 240 Bytes (!)
- Compiler für i386/Linux/ELF: 172 bytes
- Quine benötigt etwa 3 Kilobyte Größe!
- Primitive "virtuelle Maschine" mit 1 Adressregister
- 30000 Bytes Datenspeicher (am Anfang 0)
- Programmspeicher "unsichtbar"
- Varianten: (Compressed BF, Preprozessor, I/O extensions, self-modifying code...)

Brainfuck - Opcodes

- 8 Befehle:

Opcode	Bedeutung	C-Code
>	Adressregister erhöhen	++p;
<	Adressregister erniedrigen	--p;
+	Speicherzelle an aktueller Adresse erhöhen	++*p;
-	Speicherzelle an aktueller Adresse erniedrigen	--*p;
.	Aktuelle Speicherzelle ausgeben	putchar(*p);
,	Einlesen und speichern in aktueller Speicherzelle	*p=getchar()
[Springe hinter die zugehörige], falls p=0	while (*p) {
]	Springe zur zugehörigen [}

Brainfuck

- Kürzester BF-Interpreter in C:

```
char m[9999],*n[99],*r=m,*p=m+5000,**s=n,d,c;main(){for(read(0,r,4000);c=*r;r++)c-' '|| (d>1|| (r=*p?*s:(--s,r)),!d|| d--),c-' '| |d++|| (*++s=r),d|| (*p+=c=='+',*p-=c=='-',p+=c=='>',p-=c=='<',c-'.' || write(2,p,1),c-', ' || read(2,p,1));}
```

- Hello World in Brainfuck:

```
>+++++++[<+++++++&gt;-]<.>+++++++[<++++&gt;-]<+.+++++. .+++.[-]>+++++++[<++++&gt;-]  
<.>+++++++&gt;-]&lt;-.----- .+++----- .-----.[-]>+++++++[<++++&gt;-]  
<+. [-]>+++++++.
```

Brainfuck-Interpreter in Brainfuck

- Selbst-Ausführung Beweis für "Erwachsenheit" einer Sprache

```
>>>, [->+<<]>> [-<<+>>]>++++ [+++++>-] <+< [->+>>+<<<<]>>>> [-<<<<+>>>>  
>>] <<<< [->+>+<<<<]>>>> [-<<<<+>>>>] <<<< [> [->+<] < [-] ]> [-] ]<<<<->-< [->+>>>>  
<<<<]>>>> [-<<<<+>>>>] <<<< [->+>+<<<<]>>>> [-<<<<+>>>>] <<<< [> [->+<] < [-] ]> [-] ]<  
<<<< [->+<<] > [->+<] > [-] ]<<<< [->+<<]>>>> [-<<<<+>>>>] >+++++ [+++++>-] <+< [->  
>>+>+<<<<]>>>> [-<<<<+>>>>] <<<< [->+>>+<<<<]>>>> [-<<<<+>>>>] < [< [->+<<] > [-] ]<  
[-] ]>> [-] ]<<<<->-< [->+>+<<<<]>>>> [-<<<<+>>>>] <<<< [->+>>+<<<<]>>>> [-<<<<+>>>>  
>>] < [< [->+<<] > [-] ]< [-] ]>> <<<< [->+<<<<] > [->+<<] >+ [-> [-] ]< [<<<<+>>>>  
> [-] ]<<<< [->+<<<<]>>>> [-<<<<+>>>>] >+++++ [+++++>-] <+< [->+>>+<<<<]>>>> [-<<<<+>>>>]  
<<<<+>>>>] <<<< [->+>>+<<<<]>>>> [-<<<<+>>>>] < [< [->+<<] > [-] ]< [-] ]>> [-] ]<<<<->-< [->+>+<<<<]  
>>>>+<<<<]>>>> [-<<<<+>>>>] <<<< [->+>>+<<<<]>>>> [-<<<<+>>>>] < [< [->+<<] > [-] ]<  
> [-] ]>> <<<< [->+<<<<] > [->+<<] >+ [-> [-] ]< [<<<<+>>>>] > [-] ]<<<< [->+<<<<]  
>> [-<<<<+>>>>] >+++++ [+++++>-] <+< [->+>>+<<<<]>>>> [-<<<<+>>>>] <<<< [->+>>+<<<<]  
>>+<<<<]>>>> [-<<<<+>>>>] < [< [->+<<] > [-] ]< [-] ]>> [-] ]<<<<->-< [->+>+<<<<]>>>>  
> [-<<<<+>>>>] <<<< [->+>>+<<<<]>>>> [-<<<<+>>>>] < [< [->+<<] > [-] ]< [-] ]>> <<<< [->  
>>+<<<<]> [->+<<] >+ [-> [-] ]< [<<<<+>>>>] > [-] ]<<<< [->+<<<<]>>>> [-<<<<+>>>>] >+>>
```


- Arbeitet mit einem Stack of Stacks
- Angedachte Erweiterungen: Sechseckige Felder, Klein-Bottles etc. :-)

BeFunge - Beispiele

- Hello World:

```

      v
>v"Hello world!"0<
',:
^_25*,@

```

- 99 Bottles of Beer:

```

92+9*          :. v <
>v"bottles of beer on the wall"+910<
',:
^_ $          :.v
      >v"bottles of beer"+910<
      ',:
      ^_ $          v
>v"Take one down, pass it around"+910<
',:
^_ $          1-v
      :
      >v"bottles of beer"+910.:_ v
      ',:
      ^_ $          ^
      >v" no more beer..."910<
      ',:
      ^_ $$ @

```

- Quine mit 14 Bytes:

```
:0g,:93+`#@_1+
```

Java2k

- Propabilistische Programmiersprache
- IDE namens DIE verfügbar für Win32, Linux und Amiga
- Bezeichner: Vielfache von 7 zur Basis 11 (Ziffern: "0123456789 ")
- Alle Funktionen haben genau 2 Argumente
 - Ausnahme: If-Goto-Else-Statement
- Variablen nur über indirekte Arrays
- Statements haben genau 90% "Erfolgswahrscheinlichkeit"
 - Ausnahme: "119 " - Wahrscheinlichkeit erhöhen

Java2k

- Selbstdefinierte Funktionen nur indirekt aufrufbar:
 - EVALUATE-FUNCTION-BRANCH-EQUAL-CALL-OTHERWISE function

```
A<=>~(B*C);
```

- Berechne selbstdefinierte Funktion A
- Wenn A 0 zurückgibt, rufe Funktion C auf
- Andernfalls, rufe Instruktionssequenz B auf

- Zahlen nicht als Literale möglich, nur durch Errechnung
 - Hilfe: * für Zufallszahl, _ wiederholt letztes Argument
 - Erzeugen von 1 mit 90% Wahrscheinlichkeit:

```
11 6*/_ \
```

- Erzeugen von 2 mit 81% Wahrscheinlichkeit:

```
125 /11 6*/_/_ \
```

Java2k

- Einziger logischer Operator: "13 2", implementiert NAND
 - A OR B schreibbar als:

```
13 2/13 2/A/B\ /13 2/A/B \ \
```

- Whitespaces: -, A-D,F-R,T-Z
 - Können überall (auch innerhalb von Bezeichnern) sein
 - Kommentare gibt es nicht mehr ("use whitespaces instead")
 - "selten benutzte" jedoch E und S reserviert zur Thread-ID-Adressierung
 - Threads wurden im selben Release abgeschafft.

- Die Doku an sich ist genauso schlimm, wie die Sprache

- z. B. Auszug aus dem Changelog:

```
Changes from 0.0.1 to 0.0.4
```

```
Changed the last part of the version number from one-digit to two-digit format. The first digit is 4, the second the current language sub-version [stable]
```

Java2k

- (wahrscheinlich) Hello World:

```
1 1 /125 /131 /119 /125 /11 6*/_/_ \ /125 /13 2
/*/_/_ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2
/*/_/_ \ \ /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
/_ \ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ \ /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \ \
\ \ \ \ \ /*1 1 /125 /119 /11 6*/_/_ \ /13 2/*/_ \ \
125 /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
/_ \ \ /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \ \
/125 /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ /131 /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \
\_ \ \ /119 /125 /11 6*/_/_ \ /125 /13 2/*/_ \ \
\ \ \ \ \ /*1 1 /125 /131 /119 /125 /11 6*/_/_ \ \
/125 /13 2/*/_ \ \ \ /119 /125 /11 6*/_/_ \ \
125 /13 2/*/_ \ \ \ /125 /131 /119 /125 /11 6*/_/_ \
\_ \ \ /125 /13 2/*/_ \ \ \ /131 /119 /125 /11 6*/_/_ \
\_ \ \ /125 /13 2/*/_ \ \ \ /119 /125 /11 6*/_/_ \ \
/125 /13 2/*/_ \ \ \ \ /125 /131 /119 /125 /11 6/
*/_ \ \ \ /125 /13 2/*/_ \ \ \ /131 /119 /125 /11 6/
*/_ \ \ \ /125 /13 2/*/_ \ \ \ /131 /119 /125 /11 6/
*/_ \ \ \ /125 /13 2/*/_ \ \ \ /131 /119 /125 /11 6/
```



```

131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
125 /131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
/_\119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
125 /13 2/*/_\/_\125 /131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\119 /125 /11 6/*/_\/_\
125 /13 2/*/_\/_\125 /131 /119 /125 /11 6/*
*/_/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*
*/_/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*
*/_/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*
*/_/_\125 /13 2/*/_\/_\119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\125 /131 /119 /125 /11 6/*
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*
/_\125 /13 2/*/_\/_\119 /125 /11 6/*/_\/_\
/_\125 /13 2/*/_\/_\125 /13 2/*/_\/_\119 /
125 /11 6/*/_\/_\125 /13 2/*/_\/_\119 /
125 /11 6/*/_\/_\125 /13 2/*/_\/_\125 /
131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
125 /131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\
/_\119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\
/_\131 /119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\
/_\119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
/_\119 /125 /11 6/*/_\/_\125 /13 2/*/_\/_\119
125 /13 2/*/_\/_\125 /131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\131 /119 /125 /11 6/*/_\
/_\125 /13 2/*/_\/_\119 /125 /11 6/*/_\/_\
125 /13 2/*/_\/_\125 /131 /119 /125 /11 6/*/_\

```

Chef

- Programme sehen aus wie Kochrezepte
- Zutaten sind Daten (Variablen).
 - Flüssige Zutaten beinhalten Unicode-Zeichen
 - alle sonstigen Zutaten beinhalten Zahlen
 - Liquify etc. zum umwandeln
- Stacks zum Speichern: "Rührschüsseln und Backformen"

- Optionale Statements für Kochzeit und Ofen-Temperatur
- Unterprogramme:
 - hinten angehängte Zutaten-Rezepte (z. B. Soßen)
 - Haben eigene Rührschüsseln
 - Aber: auch Zugriff auf Kopien der Haupt-Behälter
 - Liefern exakt eine Rührschüssel zurück.

Chef

- Hello-World-Soufflet:

Hello World Souffle.

This recipe prints the immortal words "Hello world!", in a basically brute force way. It also makes a lot of food for one person.

Ingredients.

72 g haricot beans
 101 eggs
 108 g lard
 111 cups oil
 32 zucchinis
 119 ml water
 114 g red salmon
 100 g dijon mustard
 33 potatoes

Method.

Put potatoes into the mixing bowl. Put dijon mustard into the mixing bowl. Put lard into the mixing bowl. Put red salmon into the mixing bowl. Put oil into the mixing bowl. Put water into the mixing bowl. Put zucchinis into the mixing bowl. Put oil into the mixing bowl. Put lard into the mixing bowl. Put lard into the mixing bowl. Put eggs into the mixing bowl. Put haricot beans into the mixing bowl. Liquify contents of the mixing bowl. Pour contents of the mixing bowl into the baking dish.

Serves 1.

Beatnick

- Einfach zu lernen:
 - Sehr wenige Kommandos
 - Sehr freie Syntax
 - Vokabel-Referenz in jedem Spielzeuggeschäft erhältlich
- Bedeutung der Wörter errechnet sich aus Scrabble-Punktzahl
- Stack-basierte Maschine
- Auszug aus der Befehlstabelle:

Score	Function
<5	Does nothing. The Beatnik Interpreter may mock you for your poor scoring, at its discretion. Low scoring words such as "I" or "of" are probably not good words to program with immediately after stealing all of the interpreter's cigarettes and stomping on its beret.
5	Finds the score of the <i>next</i> word and pushes it onto the stack. Skips the aforementioned next word.
13	Pop a number from the stack, and figure out the score of the next word. If the number from the stack is zero, skip ahead by n words, where n is the score of the next word. (The skipping is actually n+1 words, because the word scored to give us n is also skipped.)
17	Stop the program.
18-23	Does nothing. However, the score is high enough that the Beatnik Interpreter will not mock you, unless it's had a really bad day.
>23	Garners "Beatnik applause" for the programmer. This generally consists of reserved finger-snapping.

Beatnick

- Wieso heißt das denn Beatnick?
- Antwort des Autors ist das folgende Beispielprogramm (gibt "Hi" aus)

```
Baa, badassed areas!  
Jarheads' arses  
    queasy nude adverbs!  
    Dare address abase adder? *bares baser dadas* HA!  
Equalize, add bezique, bra emblaze.  
    He (quezal), aeons liable. Label lilac "bulla," ocean sauce!  
Ends, addends,  
    duodena sounded amends.
```

- Noch Fragen, Kienzle?

Obfuscated Programming

- Die Kunst, Programme so zu schreiben, daß kein Mensch den Quelltext wieder verstehen kann.
- Wird insbesondere von Perl-Programmierern immer wieder gerne gepflegt :-)

Perl-Camel

- Programm-Quelltext:

```
#!/usr/bin/perl -w  
use strict;  
  
        $_='ev  
        al("seek\040D  
ATA,0,  
    {<DATA>};my  
my$Camel ;while(  
9s",$_);my@dromedary  
_=<DATA>){@camellhum  
ry1){my$camellhump=0  
    t(@dromedary1  
    $CAMEL--;if(d  
    $camellhump+=1  
@camellhump)&&/\S/){$camellhump+=1<<$CAMEL;}$CAMEL--;if(  
defined($_=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=$_;tr/LJF7\173\175\047/\061\062\063  
45678;/tr/12345678/JL7F\175\173\047`/;$_=reverse;print"$_\040  
$Camel\\n";}foreach(@camellhump){chomp;$Camel=$_;y/LJF7\173\17  
5\047/12345678;/tr/12345678/JL7F\175\173\047`/;$_=reverse;p  
rint"\040$_$Camel\\n";}#japh-Erudil';;s;\s*;;g;eval; eval  
("seek\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\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\  
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

```

```

# camel code, copywrite 2000 by Stephen B. Jenkins
# The use of a camel image with the topic of Perl
# is a trademark of O'Reilly & Associates, Inc.
# Used with permission.

```

Perl-Camel

- Ausgabe des Programms:

```

mJXXLm.          .mJXXLm
JXXXXXXXXXL.      JXXLm.      .mJXXL      .JXXXXXXXXXL
{XXXXXXXXXXXX.    JXXmXXXXm  mXXXXmXXXXL .XXXXXXXXXXXX}
.XXXXXXXXXXXXXXL. {XXXXXXXXXXXXF 7XXXXXXXXXX} .JXXXXXXXXXXXXX.
JXXXXXXXXXXXXXXXXXL. `XXXXX.      .XXXXX'.JXXXXXXXXXXXXXXXXXL
JXXXXXXXXXXXXXXXXXXXXXXXXXmXXXXXXXX. .XXXXXmXXXXXXXXXXXXXXXXXXXXXL
.XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX. {XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
.XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
JXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXFF 7XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXL
XX^7XXXXXXXXXXXXXXXXXXXXXXXXXXXXF    7XXXXXXXXXXXXXXXXXXXXXXXXXXXXF^XX
XX {XXXFXXXXXXXXXXXXXXXXXXXXXF'     `7XXXXXXXXXXXXXXXXXXXX7XXX} XX
`X}{XXX'7XXXFXXXXX^XXXX ` `      ` ` XXXX^XXXX7XXXFX`XXX}{X'
`XXX' {XXX'XXXX 7XXXFX           7XXXFX XXXX`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 7XXXFX `XX                  XX' 7XXXFX  XX
XX. JXXX. 7X.                  .XF .XXXXL .XX
{XXL 7XF7XXX. {XX              XX} .XXXFX7XF JXX}
`XXX' `XXXm                    mXXX' `XXX'

mmm.m.mmm..mm7XXXX.mm.mm. .mm.mm.XXXXFmm..mmm.m.mmm
`7LFJXXJX\FLXXL7FLLXJF\X{ }X\7LXJJ7FJXXJ7\XLXXL7JF'
`^^^^^`^`^^^^^`

.mJXXLm          mJXXLm.
.mJXXL      .JXXXXXXXXXL      JXXXXXXXXXL.      JXXLm.
mXXXXmXXXXL .XXXXXXXXXXXX}      {XXXXXXXXXXXX.      JXXmXXXXm
7XXXXXXXXXX} .JXXXXXXXXXXXXX.      .XXXXXXXXXXXXXL. {XXXXXXXXXXXXF
.XXXXXX'.JXXXXXXXXXXXXXXXXXL      JXXXXXXXXXXXXXXXXXL.`XXXXX.
.XXXXXXmXXXXXXXXXXXXXXXXXXXXXL      JXXXXXXXXXXXXXXXXXXXXmXXXXXXXX.
{XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.      .XXXXXXXXXXXXXXXXXXXXXXXXXXXXX}
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX.      .XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXXXXXXXXXXXXXXXX      JXXXXXXXXXXXXXXXXXXXXXXXXXXXXF
7XXXXXXXXXXXXXXXXXXXXF^XX          XX^7XXXXXXXXXXXXXXXXXXXXF
`7XXXXXXXXXXXXXXXXXXXX7XXX} XX    XX {XXXFXXXXXXXXXXXXXF'
` ` XXXX^XXXX7XXXFX`XXX}{X'      `X}{XXX'7XXXFXXXXX^XXXX ` `
7XXXFX XXXX`XXX} `XXX` `      ` `XXX' {XXX'XXXX 7XXXFX
{XXX' {XXXX} 7XX} {XX.          .XX} {XXF {XXXX} `XXX}
{XXF {XXF' JXX'  XX}          {XX `XXL `7XX} 7XX}
XX} 7XXm JXX'  XX'          `XX `XXL mXXF {XX
XX' 7XXXFX  XX            XX 7XXXFX `XX
.XF .XXXXL .XX            XX. JXXX. 7X.
XX} .XXXFX7XF JXX}       {XXL 7XF7XXX. {XX
mXXX' `XXX'              `XXX' `XXXm

.mmm.mm.XXXXFmm..mmm.m.mmm      mmm.m.mmm..mm7XXXX.mm.mm.
}X\7LXJJ7FJXXJ7\XLXXL7JF'      `7LFJXXJX\FLXXL7FLLXJF\X{
^^^^^`^`^^^^^`                  `^^^^^`^`^^^^^`

```

The use of a camel image in association with Perl is a trademark of O'Reilly & Associates, Inc. Used with permission.#camelcode,copywrite2000byStephenB.Jenkins

IOCCC

- The International Obfuscated C Code Contest
- <http://www.ioccc.org/>
- (fast) jährlicher Wettbewerb

IOCCC - Appetithäppchen 2001

hello_2001.c

```
#include <stdio.h>
#define S(s)char x[]=#s;s
#define Q(x)x
#define A(x,y)y##x
#define B(x,y)A(y,x)
#define C(x,y)B(y,x)
#define Z(s,t,u)case s:if(*p!=32){t;}else{u;}break;
S(B( A( a ,m ),A(n ,i)))( {B( A(h,c ),A(r ,a ))*p=x ;B(A( n, i),t)t
=0;B(A(n , i),t)s =0;B( f ,A(r, o )) (;*p;Q( p++)){C( B( A(c,t) ,h),B(A(
w, s),i))( s){ Z( 0,t+=8 *8-00 ,s ++)}Z( 1,t+= 8 ;,s++ )Z
( 2, t++ ,putchar(t-73);t=s=0)}}}
```

Ausgabe:

Hello, world!

IOCCC - anderson.c

- Contest 2000 - anderson.c
- Konvertiert stdin in ASCII-Grafik Flaggencode
- Beispiel: Hello World als Eingabe erzeugt die Ausgabe

```
<> <> <> <> <> <> <>
_()  ()/  ()/  ()/  _\  ()  (/  \
[]/^  |^^  /^^  /^^  []^^  |^^  ^^  []  []^^
<>[  [][  <>][  <>][  ][  ][  ][  ][
```

```
_()_  ()/  []  ()  ()
[] ^^ []  /^^  |  |  |
][  <>][  ][  ][  ][  ][
```

IOCCC - anderson.c

```
/* anderson.c */
#include <stdio.h>

char
*T="IeJKLMaYQCE]jbZRsKc[SldU^v\X\\|/_<[<:90!\ "$434-./2>]s",
K[3][1000],*F,x,A,*M[2],*J,r[4],*g,N,Y,*Q,W,*k,q,D;X(){r [r
[r[3]=M[1-(x&1)][*r=W,1],2]=*Q+2,1]=x+1+Y,*g+=(((x&
-1)>>1)-1)?*r:r[x>>3],(++x<*r)&&X();}E(){A|X(x=0,g
),x=7&(*T>>A*3),J[(x[F]-W-x)^A*7]=Q[x&3]^A*( *M)[2
x&1],g=J+(x[k]-W)^A*7)-A,g[1]=( *M)[ *g=M[T+=A
][x&1],x&1],(A^=1)&&(E(),J+=W);}l(){E(--q&&1
);}B(){*J&&B((D=*J,Q[2]<D&&D<k[1]&&(*g+=1
)!(D-W&&D-9&&D-10&&D-13)&&(!*r&&(*g+=0)
,r=1)||64<D&&D<91&&(*r=0,*g+=D-63)||D
97&&D<123&&(*r=0,*g+=D-95)||!(D-k[
])&&(*r=0,*g+=12)||D>k[3]&&D<=k[
-1&&(*r=0,*g+=D-47),J++);}j(
putchar(A);}b(){(j(A=( *K)[D*
r[2]*Y+x]),++x<Y)&&b();}t
{(j((b(D=q[g],x=0),A=W)
++q<*(r+1)<Y?*r+1):
)&&t();}R(){(A=(t(
0),' \n'),j(),++r
[2
```



```

]<N)&&R();}O()
j((r[2]=0,R(
),r[1]-=q)
O(g--=q)
C){(
gets
[1]))&&C((B(g=K[2]),*r=!(!*r&&>(*g++=0)),(*r)[r]=g-K[2],g=K[2
],r[
1]&&
O())
);;}
main
){C
((l(
(J=(
A=0)
[K],
A[M]
=(F=
(k=(
M[!A
]=Q
=T+(
q=(Y
=(W=
32)-
(N=4
))))
+N)+
2)+7
)+7)
),Y=
N<<(
*r=!
-A)
);;}

```

IOCCC - dhyang.c

- dhyang.c - autor: Don Yang
- saitou.c ist ein Bild von Saitou Hajime
- Es generiert einen Quine-Dreierzyklus, die Bilder Saitous Motto symbolisieren:
 - aku (sin - Sünde)
 - soku (swift - schnell, geschickt)
 - zan (slay - töten)

IOCCC - dhyang.c

```

#define/**/X
char*d="X0[!4cM,! "
"4cK`*!4cJc(!4cHg&!4c$`j"
"8f'!&~]9e)!'|:d+)rAc-!*m*"
":d/!4c(b4e0!1r2e2!/t0e4!-y-c6!"
"+|,c6!)f$b(h*c6!(d'b(i)d5!(b*a'`&c"
")c5!'b+`&b'c)c4!&b-_$c'd*c3!&a.h'd+"
"d1!%a/g'e+e0!%b-g(d.d/!&c*h'd1d-!(d%g)"
"d4d+!*1,d7d)!,h-d;c'!.b0c>d%!A`Dc$![7]35E"
"! '1cA,,!2kE`*!-s@d(!k(f//g&!f.e5'f(!+a+)"
"f%2g*!?!f5f,!=-*e/!<d6e1!9e0'f3!6f)-g5!4d*b"
"+e6!0f%k)d7!+~^'c7!)z/d-+!'n%a0(d5!%c1a+/d4"
"!2)c9e2!9b;e1!8b>e/! 7cAd-!5fAe+!7fBe(!"
"8hBd&! :iAd$![7S,Q0!1 bF 7!1b?'_6!1c,8b4"
"!2b*a,*d3!2n4f2!${4 f. '!%y4e5!&f%"
"d-^-d7!4c+b)d9!4c-a 'd :!/i('`&d"
";!+1'a+d<!)l*b(d=! ' m- a &d>!&d'"
"`0_&c?!$dAc@!$cBc@!$ b < ^&d$`"
":!$d9_&l1++^$!%f3a' n1 _ $ !&"

```

```

"f/c(o/_%!(f+c)q*c      %!      *      f &d+"
"f$S&!-n,d)n(!0i-      c-      k)      ! 3d"
"/b0h*!H`7a,!{7*      i]      5      4 71"
"[=ohr&o*t*q*`d      *v      *r      ; 02"
"7*~h./}tcrsth      &t      :      r 9b"
"].,b-725-.t--//      #r      [      < t8-"
"752793? <.~;b      ].t--r      /      # 53"
"7-r[/9~X .v90      <6/<.v;-52/{      k      goh"
"../}q; u vto      hr `i*$engt$      $      ,b"
";$/      =t ;v;      6      =`it.`;7=`      :      ,b-"
"725      = / o`.      .d      ;b]`--[/+      55/      }o"
"`d : - ?5      /      }o`. '      v/i]q      - "
"-[; 5 2 =`      it      .      o;53-      . "
"v96 <7 /      =o      :      d      =o"
"---/i ]q--      [;      h.      /      = "
"i]q--[ ;v      9h      ./      <      - "
"52={cj u      c&`      i t      . o      ; "
"?4=o:d=      o--      / i      ]q      - "
"-[;54={ cj      uc&      i]q      -      -"
"[;76=i]q[;6      =vsr      u.i      /      ={"
"=),BihY_gha      ,)\0      "      ,      o [
3217];int i,      r,w,f      ,      b      ,x ,
p;n(){return      r <X      X      X X
768?d[X(143+      X r++      +      *d      ) %
768]:r>2659      ? 59:      (      x      = d
[(r+-768)%      X 947      +      768]      ) ?
x^(p?6:0):(p      = 34      X      X      X )
;}s(){for(x=      n      )      (      x^      ( p
?6:0))=32;x=      n      )      ;return x      ; }
void/**/main X      ()      {      r      = p
=0;w=sprintf (X      X      X      X X      X o
,"char*d="); for      (      f=1;f <      * d
+143);if(33-(      b=d      [      f++ X      ] )
){if(b<93){if      X(!      p      )      o
[w++] =34;for      X(i      =      35      +
(p?0:1);i<b;      i++      )      o
[w++] =s();o[      w++      ]
=p?s():34;}      else      X
{for(i=92;      i<b;      i
++)o[w++] =      32;}      }
else o      [w++      ]
=10;o      [
w]=0      ;
puts(o);}

```

IOCCC - dhyang.c - aku.c

```

char*d=")35E!'1cA,,!" "2kE`*!-s@d(!k(f//g&! )f.e5'f(!+a+)f%2g*! ?f5f,!=f-*e/!<d"
"6e1!9e0' f3!6f)-g" "5!4d*b+e6!0f%k)d7!+~^c7!)z/d-+!'n%a0(d5!" "%c1a+/d4"
"!2)c9e2!9b;e1!8b" ">e/!7cAd-!5fAe+!7fBe(!8hBd&! :iAd$![7S,Q" "0!1bF7"
"!1b?`_6!1c,8b4!2" "b*a,*d3!2n4f2!${4f.'!%y4e5!&f% d-^-d7!4" "c+b)"
"d9!4c-a'd:!/i('`" "&d;!+l'a+d<!)l*b(d=! 'm-a&d>!&d' `0_&c" "?!"
"$dAc@!$cBc@!$b<^" "" "&d$` :!$d9_&l++^$!%f3" "a'n"
"l_" "$!&f/c(o/_%!(f+c)q*c%" "!*f&d"
"+f$" "s&!-n,d)n(!0i-c-k)!3d/" "b0h*!H`"
"7a,![" "7X0[!4cM,!4cK`*!4cJc(!4cHg&!4" "c$j8f'!&~"
"]9e)!'" "|:d+!)" "rAc-!*m*:d/!4c(b4e0!1r" "2e2!/t0e4!-"
"y-c6!+|,c6!)f$b(" "h*c6" "!(d'b(i)d5!(b*a'" "&c)c5!'b+`&"
"b'c)c4!&b-_$c" "'d*c3!&a.h'd+d" "1!%a/g'e+e0!%b"
"-g(d.d/!&c*" "h'd1d-!(d%g)" "d4d+!*l,d7d)! ,h-"
"d;c'!.b0c" ">d%!A`Dc$" "![7*i]5471[=ohr&o*"
"t*q*`d" "*v*r;027" "*~h./}tcrsth&t:r9"
"b].,b" "" "-725" "-.t--/" "/#r[<t8-752793?<.~"
";b]." "t--" "+r/#" "537-r" "[/9~X.v90<6/<.v;- "
"52/=" "{kgoh." "/}q" ";u" "vtohr" "`i*$engt$$,b;$/="
"t;v" ";6=`it." "`;" "7=`" ":,b-7" "25=/o`.d;b]`--[["
"/+" "55/}o`.d:" "" "-?5" "/}o`. ' " "v/i]q--[;52=`it"
".o" ";53-.v96<7" "/=o" ":d=o--/" "i]q--[;h./=i]"
"q" "--[;v9h./<- " "52=" "{cjuc&`" "it.o;?4=o:d="
"o" "--/i]q--[ " ";54=" "{cjuc&i]q-" "-[;76=i]q[; "
"6=" "vsru.i" "/={ " "=),BihY_gha,)" "" ,o[3217];
int i,      r,w      ,f,b,p,      t=641,x;n(){return      r<t?d[(*
d+143+(r      ++)%t]:r>      +143!59:(x=d[(r++-t)      %351+t]

```

```
?x^(p?6:0):(p+=34);} main(){w=sprintf(o,"char" *d=");r=p=0;for(f=1;f<*d+143;) if((b=d[f++])-33){if(b<+93){ if(!p)o[w++]=34;for(i=35+(p?0:1);i <b;i++)o[w++]=n();o[w++]=p?n():+34; }else for(i=92;i<b;i++)o[w++]=32;}else o[w++]=10;o[w]=0;puts(o);};/*Don_Yang*/;
```

IOCCC - dhyang.c - soku.c

```
char*d="S,Q0!1bF7!1b?'_ "6!1c,8b4!2b*a,*d3"!12n4f2!${4f.'!%y4e5!&f%d-^-d7!4c+" "b)d""9!4c-a'd:!/i(" " `&d;+!1'a+d<! )l*b(d!='m-a&d>" " !&d`0_&" "c?!$dAc@" " !&cBc@!$b<^&d$" " `: !&d9_&l++^$!%f3a'n1_!&f/c(o/_%" " ! (f+c)" "q*c%!*f&d" " +f$s&!-n,d)n(!0i-c-k)!3d/b0h" " *!H`" "7a,!" " [7X0" " [!4cM,!4cK`"*!4cJc(!4cH" " g&" " !4c$j" " 8f'!&~]9e)" " '! | :d+)rAc-!*m*: "d/!" " 4c(b" "4e0!1r2" " e2!/t0e""4!-y-" " "c""6!+|,c6!)f$b(h" " *c6!(d" " 'b(i)d5!(b*a`&c)c5!'b+`&b'" " "c)c4!&b-_&c'd*c3!" " "&a.h+d+d" "1!%a/g'e+e0!%b-g(d.d/!&c*" " "h'd1d-!(d""%g)d4d" " +! *l,d7d)!, " "h-d;c'!.b0c>d%!A`Dc$![7]" " 35E!'1cA,,!2kE`*!-" " s@d(! (k(f//g&" "!)f.e5'f(!+a+)f%2g*!?" " f5f,!f-*e/!" "<d6" " e1!9e0'f3!6f)-g" "5!4d*b+e6!0f%k)d7!" " +~^`c""7!)z/d-+!" " 'n%a0(d5!%c1a+/d4" " !2)c9e2!9b;e1!8b" " >e/!7c" " Ad-!5fA" " e+!7fBe(!8hBd&! :iA" "d$![7*i]5471" " [" " =ohr&" " o*t*q*`*d*v*r;027*~" "h./}tc" " "rst" " h&t:r9b].,b-725-.t-" " -//#r" " "[<t8-752793" " ?<.~;b].t""--r/#5" "37-" " "r" " [/9~X.v90<6/" "<.v;" " -52/={kgoh./}q;uv" "t" " ohr`.i*$engt$" "$,b;$/="t;v;6=`it.`" " ;7=`:,b-725=/o`." ".d;b]`--[/+55/" " }o`.d" " :~?5/}o`. 'v/i]q--[;52" " ="it.o;53-.v96" "<7/=o:d=o--/i]q--[;h." " /=i]q--[;v9h./<-52={cju" " c&`it.o;?4=o:" "d=o--/i]q--[;54={cju" " c&i]q--[;76=i]q[;6=vsru.i/" " ={=),BihY_g" "ha,)",o[3217];int i, r,w,f,b,p,t=641,x;n(){return r< t?d[(* d+143+(r ++ )%t]:r>+ 1341 ?59:(x=d[(r++-t)%351+t])?x^(p?6:0):(p+=34);}main(){w=sprintf(o "char"*d=");r=p=0;for(f=1;f<*d +143;) if((b=d[f++])-33){if(b <+93){if(!p)o[w++]=34;for(i=35+( p?0: 1);i<b;i++)o[w++]=n();o[ w++] =p?n():+34;}else for(i=92;i <b ;i++)o[w++]=32;}else o[w++]=10;o[w]=0;puts(o);};/*Don_Yang*/;;;;;;;;;;;
```

IOCCC - dhyang.c - zan.c

```
char*d="X0[!4cM," " !4cK`*!4cJc(!4cHg&!4c$j8f'!&~]9e)!'|:d+)rAc-""!*m*:d/!4c(b" "4e0!1r2e2!/t0" " e4!-y-c6!+|,c6!)f$b(h*c6!(d'b(i)d5""!(b*a`&c)c5!'b+`&b" " 'c)c4!&b-_&c'" " d*c3!&a.h'd+d1!%a/g'e+e0!%b"-g(" " d.d/!&c*h'd1d-!(d%" "g)d4d+!*l,d7d" " )!,h-d;c""!'.b0c>d%!A`Dc$![7]3" " 5E!'1cA,,!2kE`*!" "s@d(! (k(f//g&" "!)f.e5" " 'f(!+a+)f%2g*!" " ?f5f,!f-*e/!<d" "6e1!9e0'f3!6f)" " -g5!4d*b+e6!0f%k" " )d7!+~^`c7!)z/" " " "d-+!'n%a0(d5!%c1" " a+/d4!2)c9""e2!" "9" " b;e1!8b>e/!7cAd-" " !5fAe+!7fBe(!8hBd" "&!" " : " iAd$![7S," " Q0!1bF7!1" " b?'_6!1c,8b4!2b*a,*" "d3!2n4f2!${4f.'!" " %y4e5!&" " f%d-^- " -d7!4c+b)d9!4c-a'd:!/ "i('`&d;+!1'a+d<!" " )l*b(d='m-a" " &d>!&d`0_&c?!$dAc@!$c" "Bc@!$b<^&d$" " `: !&$""d9_ " "&l" " ++^$!%f3a'n1_!&f/c(o/_ " %!(f+c)" " q*c" " %!*f&d+" " f$s&!-n,d)n(!0i-c-k)!3d/" "b0h*!" " H`7a,!" " [7*i" " ]5471[=ohr&o*t*q*`*d*v*r;" "027" " *~=h./}tc" " rs" " th&t:r9b].,b-725-.t--//#r[" "<t" " 8-7" " 52793?<.~;b]" " .t" " --r/#537-r[/9~X.v90<6/<.v;" " " -52/={kgoh./}q;uvtohr`.i*$eng" " t$$$ ,b;$/=t;v;6=`it.`;7=`:,b- " " 725=/o`.d;b]`--[/+55/}o`.d:~?" " 5/}o`. 'v/i]q--[;52=`it.o;53-" " .v96<7/=o:d=o--/i]q--[;h" " " =i]q--[;v9h./<-52={cju" " " c&`it.o;?4=o:d=o--/i]" " q-" " -[;54={""cju&i]" " "q" " --[;76=i]q[;6=v" " sru" " .i/{=),BihY_" " "gh" " a,)",o[3217] ;int i ,r,w,f,b,p,t= 641 ,x;n() {return r <t?d[(* d+143+(r ++ )%t]:r>+ 1341 ?59:(x=d[(r ++-t )%351+t])?x^(p?6:0):( p+=34);}main() {w=sprintf(o, "char"*d=");r=p=0;for(f=1;f<*d+143;)if((b=d[f ++])-33){if(b <+93){if(!p)o[ w++] =34; for(i=35+(p?0:1);i<b;i++)o[w++]=n();o[ w++] =p?n():+34;}else for(i=92;i <b ;i++)o[w++]=32;}else o[w++]=10;o[w]=0;puts(o);};/*Don_Yang*/;;;;;;;;;;;
```

IOCCC - heathbar.c


```

MAIn      MAiN      maIn
mAlN      main      MAIn
Ma1N      MAiN      maln
mAlN      main    malN MAIn
mAiN      MAIn
Ma1N      maln
MAiN      main
mAlN      main
MALn      mAin
ma1N      maIn
ma1N      maIn
Main      MAIn

```

Selbstmodifizierender Code

- Beispiel: Perl-Programm "selfgol"
- Camian:
"The only thing better than self-modifying code is code that modifies itself before it does so."
- Macht "gleich 4 Dinge auf einmal":
 - "Game of Life": selfgol -g -x=width -y=height
 - Quine: selfgol -s
 - Ein scrollendes Banner ausgeben: selfgol -d="banner"
 - Aus einem anderen Program ein Quine machen: selfgol progname
- Unter 1000 Bytes Quellcode

Selbstmodifizierender Code

- selfgol-Quelltext:

```

#!/usr/local/bin/perl -sw
$;=$/;seek+DATA,!++$/,!$;$_=<DATA>;$s&&print||$g&&do{$y=($x||=20)*($y||8);sub
i{sleep&f}sub'p{print$;x$=,join$;, $b=~/.{$x}/g}$j=$j;sub'f{pop}sub
n{substr($b,&f%$y,3)~tr,O,O,}sub'g{$f=&f-1;($w,$w,substr($b,&f,1),O)[n($f-$x)+
n($x+$f)-(substr($b,&f,1)eq+O)+n$f]||$w}$w="\40";$b=join' ',@ARGV?<>:$_,$w
x$y;$b=~s.)$&~/\w/?O:$w)ge;substr($b,$y)=q++;$g=' $i=0;$i?$b:$c=$b;
substr+$c,$i,1,q$;$g=~s?\d+?($&+1)%$y?e;$i-$y+1?eval$g:do{$i=-1;$b=$c;p;i
l}' ;sub'e{eval$g;&e}e||eval||die+No.$;
DATA
if($j){{$^W=$|;*_ =sub{$+=s=#([A-z])(.*)#=#$+1#=#g}}
@s=(q[ $_ =sprintf+pop@s,@s],q[
if($j){{$^W=$|;*_ =sub{$+=s=#([A-z])(.*)#=#$+1#=#g}} #_The_Perl_Journal_#
@s=(q[%s],q[%s])x2;%s;print"\n"x&_,$_;i$;eval}
]x2;$_ =sprintf+pop@s,@s;print"\n"x&_,$_;i$;eval}$/= $y;$"=",",;print
q<#!/usr/local/bin/perl -sw
if(!$s){>.( $_=<>).q<}else{@s=(q[printf+pop@s,@s],q[#!/usr/local/bin/perl -sw
if(!$s){>.(s%%$%%$g,tr=[===tr]=]=||&d,$_).q<}else{@s=(q[%s],q[%s])x2;%s}
]x2;printf+pop@s,@s}
>

```

Literatur (hoffentlich vollständig)

- <http://www.google.com/> - die Suchmaschine meiner Wahl
- <http://www.ioccc.org/> - International Obfuscated C Coding Contest
- <http://www.mines.edu/students/b/bolmstea/quines/> - the Quines List
- <http://www2.latech.edu/~acm/HelloWorld.shtml> - the ACM Hello World page
- <http://pauillac.inria.fr/caml/FAQ/> - the Caml FAQ
- <http://www.cliff.biffle.org/esoterica/beatnik.html> - The Beatnik Language
- <http://www.cliff.biffle.org/esoterica/hq9plus.html> - The HQ9+ programming language
- <http://internet.ls-la.net/mirrors/99bottles/> - über 200 Beispiele
- <http://www.selectorweb.com/perl.html> - selectorweb Perl seite
- <http://www.stud.uni-hannover.de/~freise/ascii/pqr/perl.txt> - U. a. das Perl-Camel

- <http://www.catseye.mb.ca/esoteric/bf/> - "die" Brainfuck Homepage
- <http://www.muppetlabs.com/~breadbox/bf/> - eine weitere Seite zu Brainfuck
- <http://koeln.ccc.de/projekte/brainfuck/index-e.html> - Brainfuck Projekt des CCC Köln
- <http://cydathria.com/bf/brainfuck.html> - Programming in Brainfuck - Introduction
- <http://www.muppetlabs.com/~breadbox/intercal-man/> - the InterCal Programming Manual
- <http://www.catseye.mb.ca/esoteric/befunge/> - Befunge Homepage
- <http://www.p-nand-q.com/java2k.htm> - Java2k Homepage
- <http://purists.org/esoteric/> - Esoteric Language Database
- <http://www.dangermouse.net/esoteric/chef.html> - The Chef Programming Language

Ende

- <http://www.ulm.ccc.de/>
- mail@ulm.ccc.de
- Montagstreff: ab ca 19:30, Cafe Einstein (Uni)
- Verwendete Software:
 - Opera
 - Amaya
 - Mozilla
 - Debian Woody Linux
 - nedit
 - Gimp
- Ab zu Parasco?

