

LINGUAGGI ESOTERICI

@EPICODE-SECURE_SENTINELS

CONTENT

- 01** COSA SONO
- 02** CENNI STORICI
- 03** TERMINOLOGIA ESOTERICA
- 04** CODICE ESOTERICO
- 05** CONCLUSIONI

COSA SONO?



Nel vasto universo della programmazione informatica, esistono linguaggi che vanno oltre i confini del normale e del pratico. Questi linguaggi, noti come "linguaggi di programmazione esoterici", sfidano le convenzioni tradizionali e spingono la creatività dei programmatore al limite.



eat();



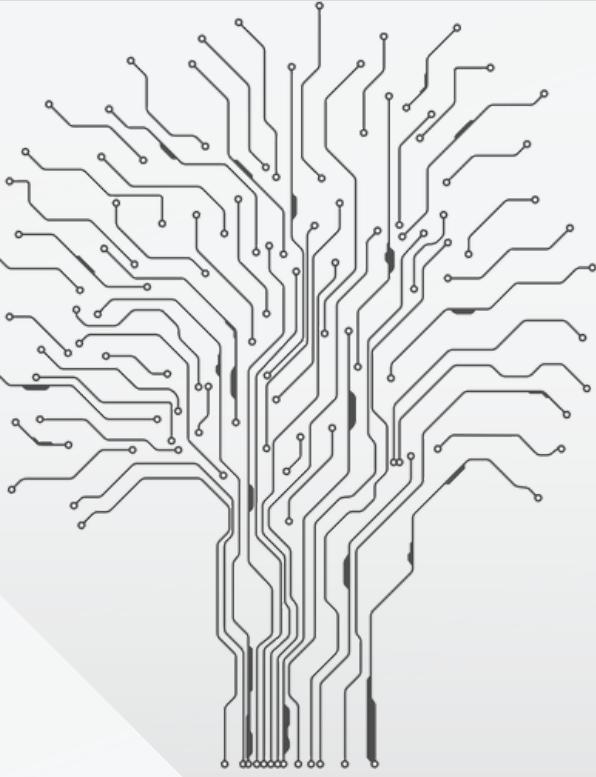
sleep();



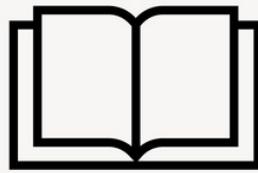
code();



repeat();

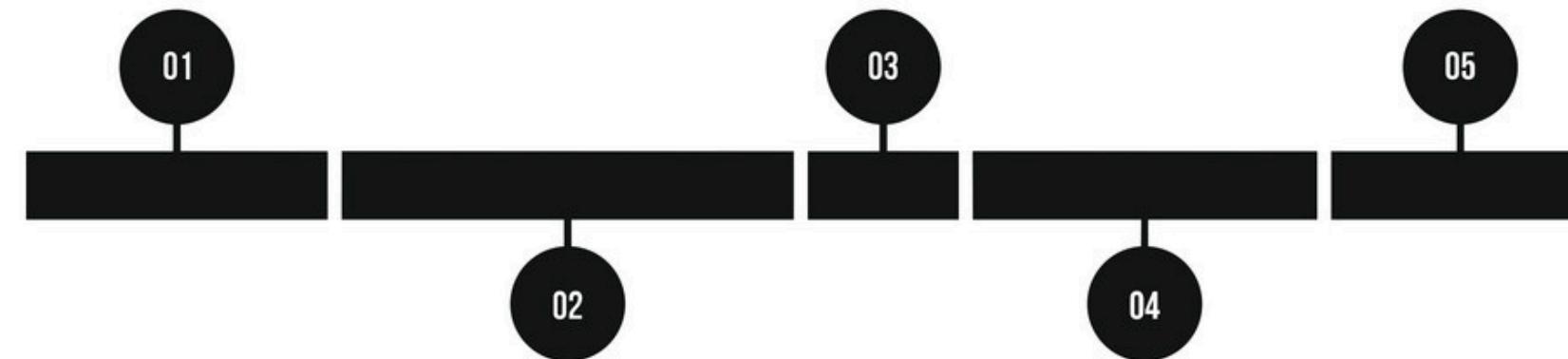
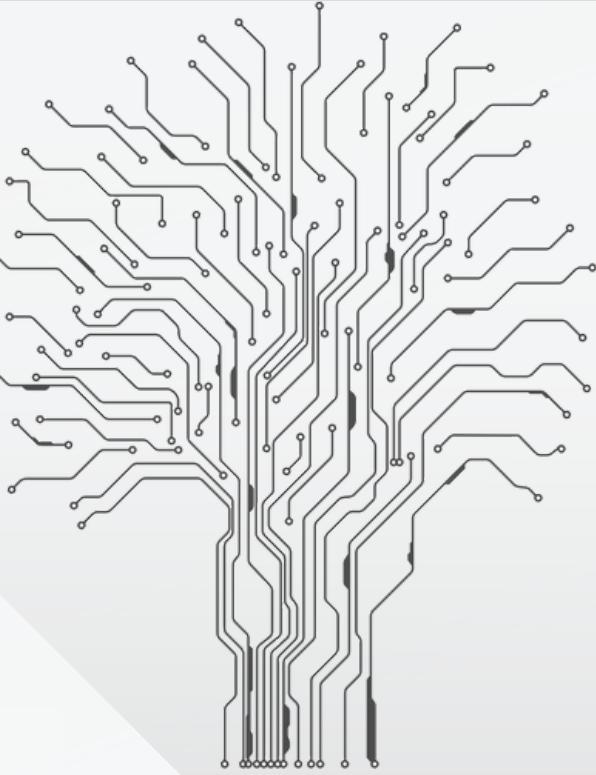


CENNI STORICI

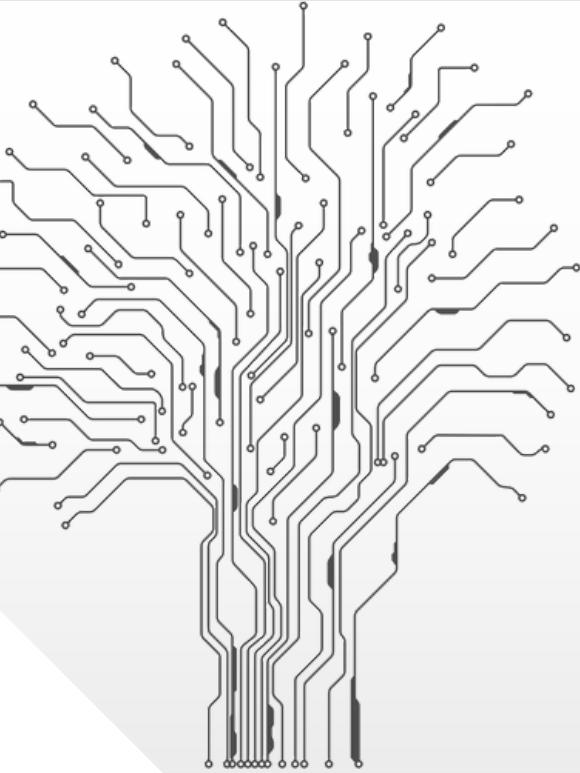


Il capostipite di questo genere di linguaggi è INTERCAL, NATO NEL 1972 DA James M. Lyon e Don Woods. La loro intenzione era quella di creare un qualcosa di totalmente nuovo e differente da ciò che già esisteva all'epoca. Effettuarono una fusione tra diversi linguaggi come ad esempio il Fortran, il COBOL, e l'assembly.

Un altro esponente degno di essere citato è il FALSE nato approssimativamente una ventina di anni dopo ad opera di Wouter Van Oortmerssen. Tra le sue peculiarità figurano: una sintassi estremamente confusa ma concisa e il concetto di macchina a stack. Il suo compilatore occupa solo 1024 byte. Un diretto discendente è il Brainfuck, che fa uso di soli otto caratteri di tipo ASCII per la stesura del codice. Ultimo ma non per importanza è il Befunge, nato grazie a Chris Pressey.



TERMINOLOGIA ESOTERICA



TURING TARPIT

Linguaggio basato su un tipo di programmazione Turing, dispone di un ristretto numero di comandi, operatori ed oggetti. Questa espressione è stata coniata da Alan Perlis nel 1982. Qualsiasi linguaggio Turing equivalente può implementare programmi calcolabili dalla macchina di Turing. I Turing Tarpit si distinguono per il fatto di offrire una macchina astratta semplice lasciando allo sviluppatore il compito di trovare la soluzione più adatta al problema da risolvere.

STATEFUL ENCODING

E' un metodo di programmazione nel quale per seguire un comando sono necessari due passaggi: serve localizzarne la posizione e poi lanciarlo. L'elenco delle operazioni può essere statico o dinamico.

FUNGE

Una cosiddetta fungo è un programma il cui codice è disposto in modo da formare una figura geometrica e la sequenza di esecuzione delle istruzioni è stabilita dal movimento di un puntatore su tale figura. Lo sviluppatore del codice ha poi vari strumenti per spostare il puntatore ed eseguire le varie porzioni di codice.

CODICE ESOTERICO



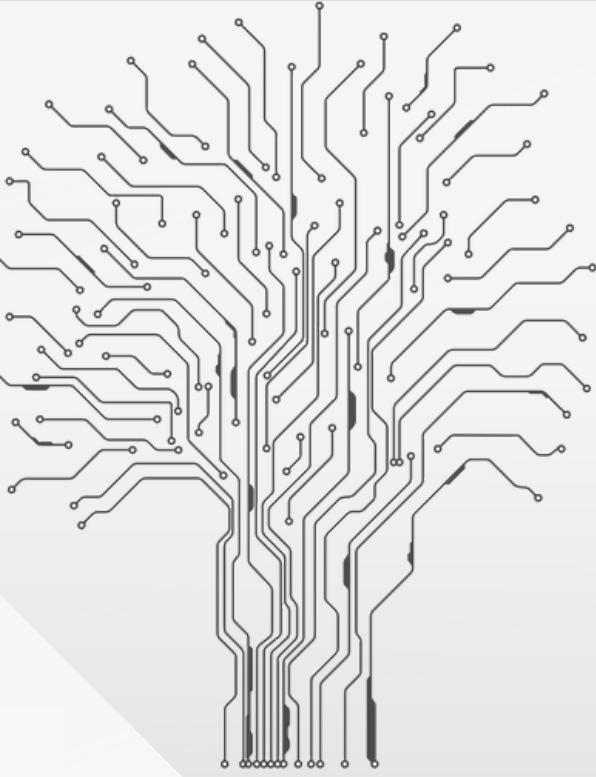
LOLCODE:

Non è così incomprensibile come gli altri ed utilizza nella sua sintassi degli slang in inglese derivanti dal fenomeno dei LOLCATS.

Esempio di codice che stampa "Hello world":

```
HAI  
CAN HAS STDIO?  
VISIBLE "HAI WORLD!"  
KTHXBYE
```

In questo esempio di codice, l'istruzione "CAN HAS" sta per "include" mentre "VISIBLE" sostituisce la "printf".



CODICE ESOTERICO



Chef:

Linguaggio esoterico basato sullo stack i cui programmi assomigliano a ricette di cucina. Alcune linee guida sono le seguenti:

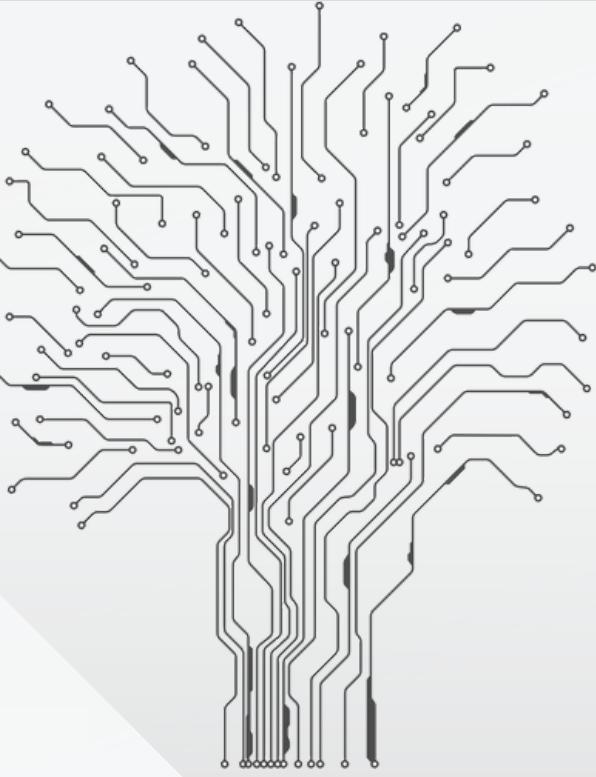
1. I programmi devono generare degli output validi rimanendo comunque deliziosi.
2. Le ricette devono andare bene per diversi budgets.
3. Oltre ad usare il sistema metrico per indicare le quantità degli ingredienti, su può approssimare usando ad esempio espressioni del tipo "un cucchiaio" o "una tazza".

Esempio (parziale) di codice che stampa "Hello world":

Hello World Souffle.

Ingredients.

*72 g haricot beans
101 eggs
108 g lard
111 cups oil
32 zucchinis
119 ml water*



CODICE ESOTERICO



Brainfuck:

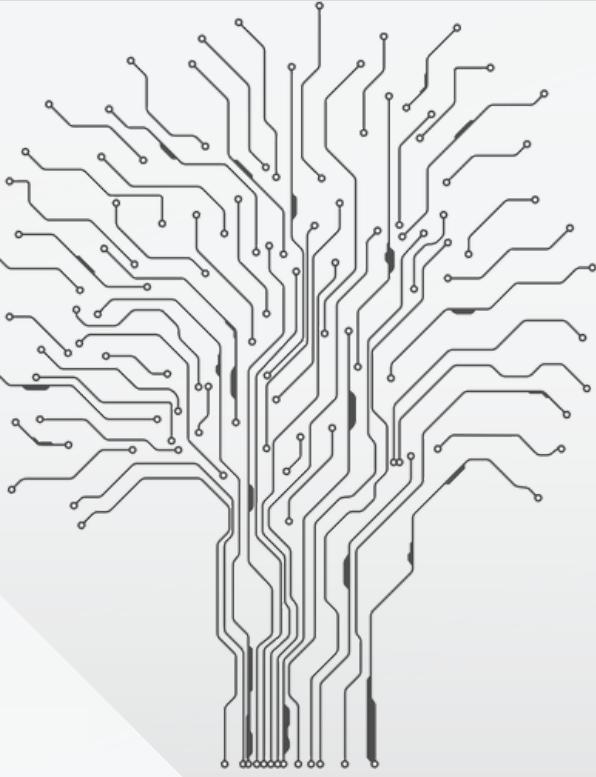
Creato da Urban Müller nel 1993. Il suo obiettivo era di scrivere un linguaggio compatibile con macchina di Turing con un compilatore più piccolo possibile. Si basa su un array di byte inizializzato a zero, un puntatore che punta al primo byte dell'array, e due stream di byte per input e output.

I sorgenti per Brainfuck possono essere transcodificati in [C](#).
Ecco le 8 istruzioni del linguaggio:

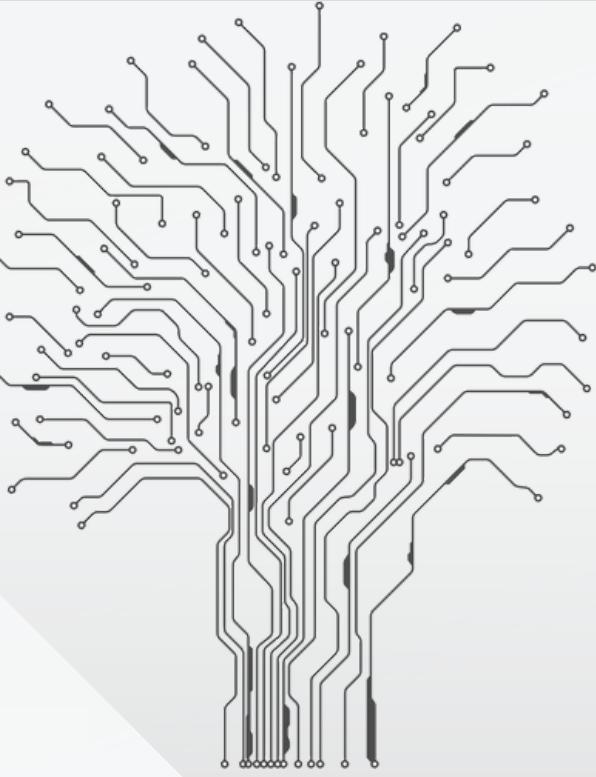
```
> incrementa il puntatore  
< decrementa il puntatore  
+ incrementa il byte indirizzato dal puntatore  
- decrementa il byte indirizzato dal puntatore  
. output dal byte indirizzato dal puntatore (ASCII)  
, input al byte indirizzato dal puntatore (ASCII)  
[ salta in avanti all'istruzione dopo il corrispondente ] se il byte indirizzato dal puntatore è zero  
] salta indietro all'istruzione dopo il corrispondente [ se il byte indirizzato dal puntatore non è zero
```

Esempio di codice scritto:

```
+++++++++[>++++++>++++++>+++>+<<<-]>++.>+.+++++.+++.>+  
.=<+++++++.>++.----.-----.>+.>.
```



CODICE ESOTERICO



Cow:

Le sue istruzioni sono tutte le possibili varianti di "moo" e possiede solo 11 istruzioni ed è un linguaggio di tipo case-sensitive. Ha un registro temporaneo in grado di contenere un intero. Si fa uso di un puntatore che inizia puntando alla prima cella ed ogni cella, prima dell'esecuzione, ha valore zero.

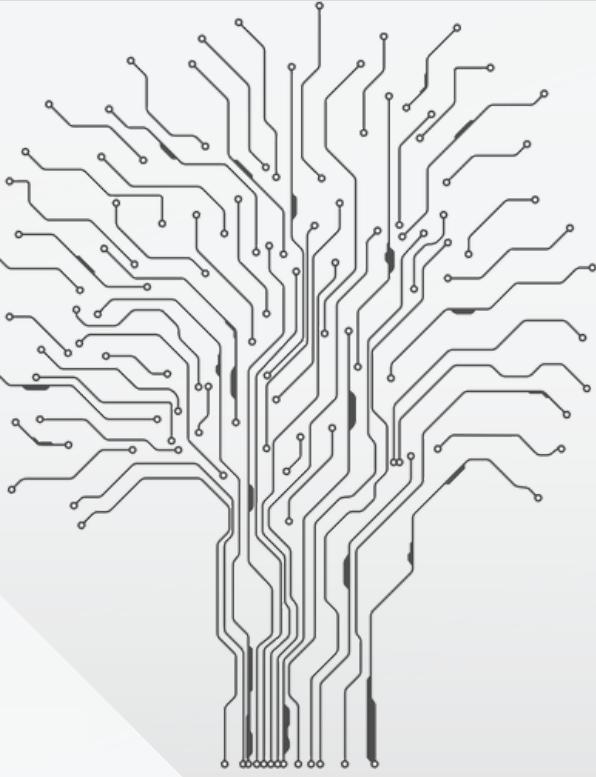
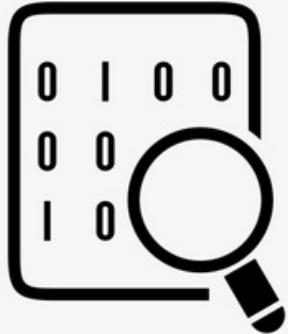
Alcune delle istruzioni:

- 1) oom: legge un intero dallo standard input e lo salva in una cella.
- 2) OOM: stampa sullo standard output i valori delle celle come interi.
- 3) MMM: si occupa di copiare il valore di una cella nel registro se quest'ultimo è vuoto. Se non è vuoto, incolla il valore nella cella corrente e svuota il registro.
- 4) 000: salva zero nella cella.
- 5) M00: torna alla chiamata effettuata da moo se la cella contiene zero. Viceversa, se la cella ha un valore diverso, continua con l'istruzione successiva.
- 6) M0o: effettua il decremento di una cella di 1.
- 7) Mo0: effettua l'incremento di una cella di 1.

Esempio di codice:

```
Moo moO MoO mOo MOO OoM MMM moO moO MMM mOo mOo moO MMM moO moO MOO MoO mOo MoO moO  
moO moO moO
```

CODICE ESOTERICO



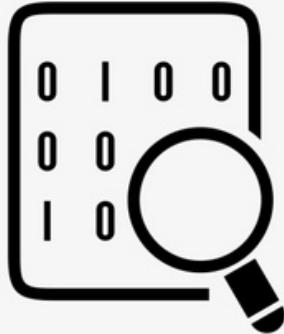
INTERCAL:

Acronimo del quale non si conosce il senso logico di "The Compiler Language With No Pronounceable Acronym". Le sue istruzioni sono davver peculiari. Eccone alcune: READ OUT, IGNORE, FORGET, PLEASE. PLEASE trova impiego all'interno del linguaggio in due modi: va utilizzato sovente per non causare un warning che avvisa che il codice scritto non è abbastanza educato; se usato in maniera sconsiderata il compilatore avviserà il programmatore di eccessiva educazione.

Esempio di codice (Hello world):

```
DO ,1 <- #13PLEASE DO ,1 SUB #1 <- #238DO ,1 SUB #2 <- #108DO ,1 SUB #3 <- #112DO ,1 SUB #4 <- #0DO  
,1 SUB #5 <- #64DO ,1 SUB #6 <- #194DO ,1 SUB #7 <- #48PLEASE DO ,1 SUB #8 <- #22DO ,1 SUB #9 <-  
#248DO ,1 SUB #10 <- #168DO ,1 SUB #11 <- #24DO ,1 SUB #12 <- #16DO ,1 SUB #13 <- #162PLEASE READ  
OUT ,1PLEASE GIVE UP
```

CODICE ESOTERICO

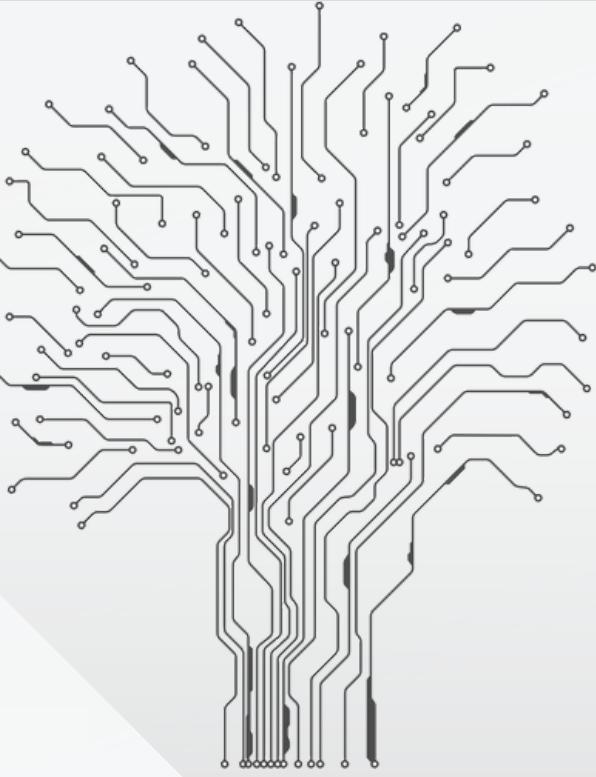


Whitespace:

Data di rilascio: 1 Aprile 2003. Non è l'ennesimo pesce d'aprile ma bensì un linguaggio che non fa uso di alcun tipo di carattere al suo interno, prediligendo invece, spazi, ritorni a capo e tabulazioni. Si potrebbe integrare del codice di questo linguaggio anche in altri programmi scritti diversamente. L'unico modo per visualizzare il codice è far clic col mouse e trascinare il puntatore nella pagina.

Spoon:

Derivato dal Brainfuck ma utilizza solo 0 e 1.



CODICE ESOTERICO



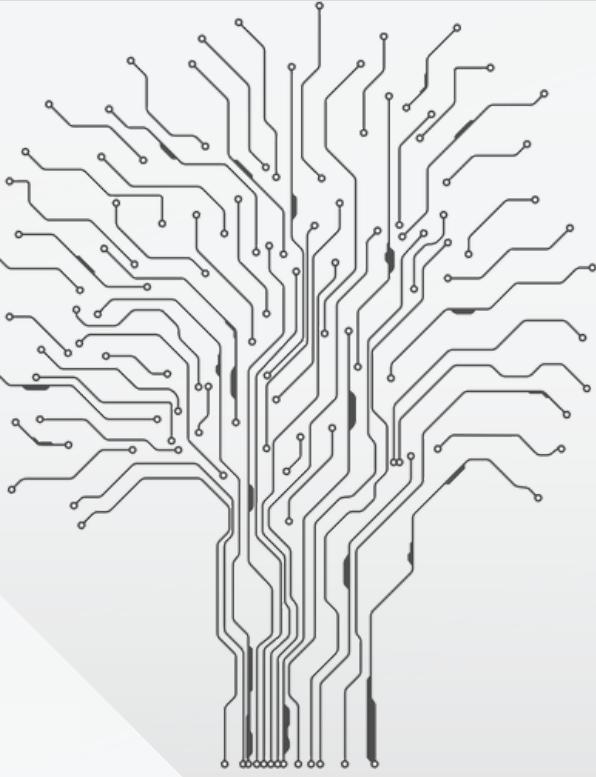
Malbogle:

Malbolge è un linguaggio di programmazione esoterico, creato da Ben Olmstead nel 1998, con lo scopo che fosse il codice più difficile possibile da utilizzare. Il nome Malbolge deriva dal nome dato all'ottavo cerchio dell'Inferno nella Divina Commedia, dove si trovano i fraudolenti.

Malbolge è stato costruito per eseguire i programmi su una macchina virtuale basata su cifre ternarie (anche dette trit). Ogni dato (word) è formato da 10 trit, con un intervallo di valori da 0 a 2222222222 in base 3, cioè 59048 in formato decimale. Anche ciascun indirizzo di memoria occupa una word. I dati e il codice del programma sono inseriti nella stessa memoria. Il programma è posto nella memoria all'avvio a partire dalla locazione 0; ogni spazio bianco è ignorato, così come ogni eventuale ritorno a capo.

Esempio di codice che stampa a schermo "Hello world!":

```
('&%:9]!~}/z2Vxwv-,P0qponl$Hjig%eB@@>}=<M:9wv6Wsu2T/nm-,jcL(I&%$#"  
'CB]V?Tx<uVtT`Rpo3NlF.Jh++FdbCBA@?]!~|4XzyTT43Qsqq(Lnmkj"fhg${z@>
```



CODICE ESOTERICO



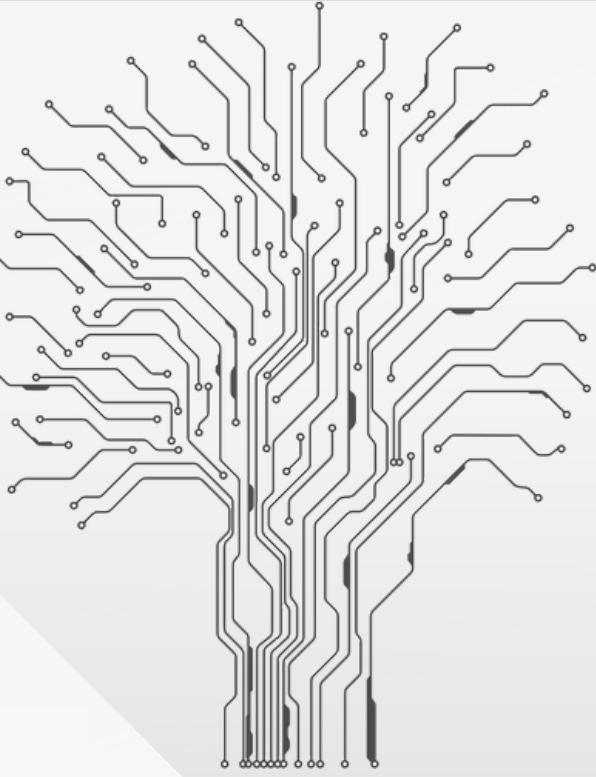
FALSE:

`FALSE` è un linguaggio di programmazione esoterico ideato da Wouter van Oortmerssen nel 1993, chiamato così in onore del valore booleano, il preferito dall'autore.

Il linguaggio è simile al Forth ma con una sintassi creata in modo da rendere il codice intrinsecamente offuscato, confuso e illeggibile. È anche noto per avere un compilatore di soli 1024 byte (scritti in 68000 Assembly). Secondo van Oortmerssen, FALSE è stato fonte di ispirazione per vari altri linguaggi esoterici molto noti, come brainfuck e Befunge.

Esempio di codice che stampa a schermo i numeri da 1 a 100:

99 9[1-\$][\\$@\$@\$@\$@\/*=[1-\$\$_[%\1-\$@]?0=[\\$.' ,\]?]?]#



CODICE ESOTERICO

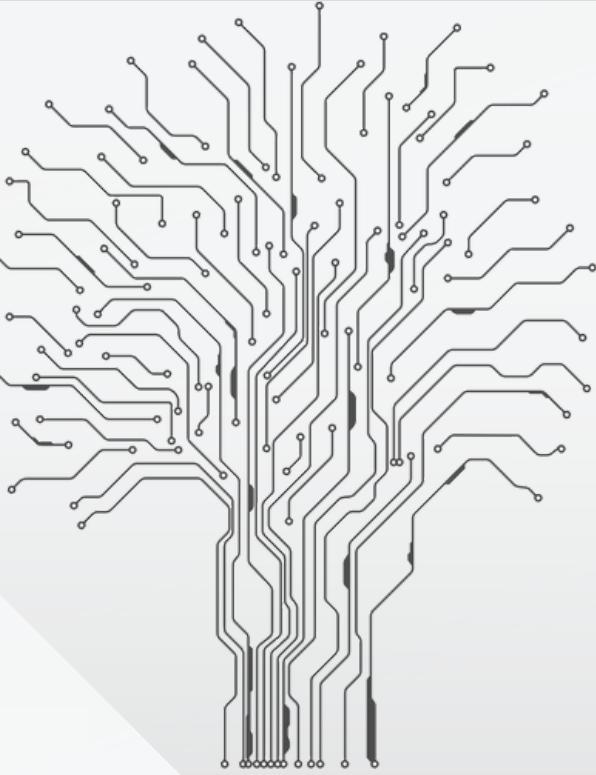


Ook!:

Molto simile al brainfuck ma la sua sintassi è costituita solo dalla parola "Ook" seguita da segni di punteggiatura come "!", "?" o ".". Secondo il suo creatore la parola "Ook" è un onomatopea che emettono gli orangotanghi coi loro versi per comunicare tra di loro. Quindi il risultante codice dovrebbe rappresentare una discussione tra questi mammiferi.

Esempio:

```
Ook. Ook? Ook.  
Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.  
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook!
```



CODICE ESOTERICO



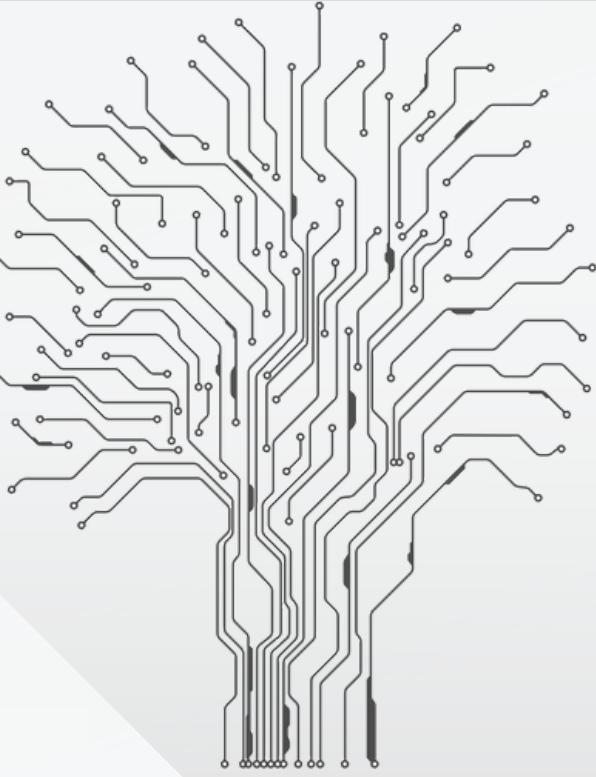
Shakespeare:

In questo linguaggio che nella sua sintassi richiama le opere dell'omonimo drammaturgo, i programmi hanno molto in comune con l'assembler, dato che seguono solo istruzioni di basso livello. I programmi risultano lunghi e poco comprensibili.

Parte del codice per stampare "Hello world":

*Romeo, a young man with a remarkable patience.
Juliet, a likewise young woman of remarkable grace.
Ophelia, a remarkable woman much in dispute with Hamlet.
Hamlet, the flatterer of Andersen Insulting A/S.*

*Act I: Hamlet's insults and flattery.
Scene I: The insulting of Romeo.
[Enter Hamlet and Romeo]*



CODICE ESOTERICO



Monicelli:

Ogni sua istruzione è una supercazzola. Si ispira al regista italiano del film "Amici miei", per l'appunto Mario Monicelli.

Ecco un esempio di programma che segue il calcolo fattoriale:

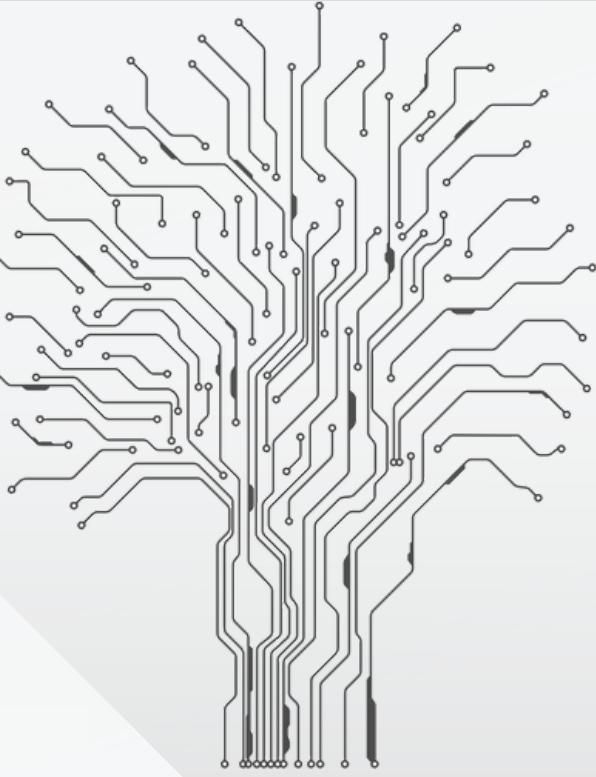
bituma un po' il calcolo del fattoriale come fosse un Sassaroli, o no?

*blinda la supercazzola Sassaroli antanizzata con pazienza bitumata, concorda
Sassaroli o scherziamo?*

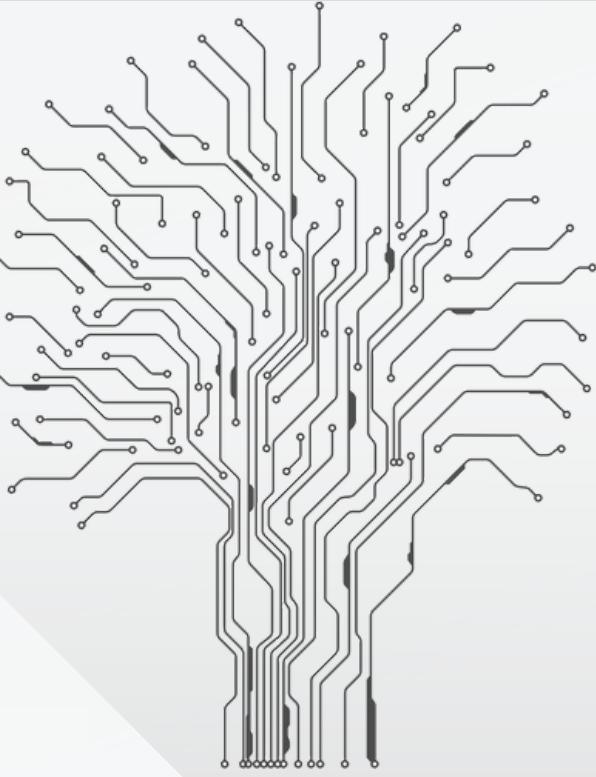
*voglio il gruppo, Sassaroli
il gruppo come fosse 1*

che cos'è la pazienza?

*1: il gruppo come fosse 2 con scappellamento a destra per 1
o tarapia tapioco: il gruppo come fosse prematurata la supercazzola
antanizzata con pazienza meno 1 o scherziamo? per la pazienza*



CONCLUSIONI



I linguaggi di programmazione esoterici sono molto più di semplici strumenti di scrittura del codice. Sono sfide intellettuali, espressioni artistiche e modi unici per spingere i confini della programmazione tradizionale. Nonostante la loro eccentricità e complessità, questi linguaggi ci ricordano che la programmazione è un'arte in continua evoluzione, dove la creatività non ha limiti e l'unicità è celebrata.

Ci sono poi anche altri linguaggi che non vengono inclusi in questa categoria, però sono dei linguaggi che anziché utilizzare la sintassi in lingua inglese alla quale tutti siamo avvezzi, usano lingue differenti come l'arabo (Jeem), l'ebraico (HPL), il russo (Rapira), il giapponese (Nadesiko), l'indiano (Hindawi) e il persiano (Farsi.NET).



Alessio
Forli



Simone
Esposito



Alejandro
Cristino



Simone
La Porta



Nicolo
Callegaro



Grazia
Coco



Gianluca
Sansone



SECURE SENTINELS