

I recommend that you view it as a pdf and dark mode(for font and image): <https://glki.netlify.app/nl.pdf>

official youtube channel(for emergency): <https://youtube.com/@normallanguage?si=4E60EKgSuZa9OJAj>

all feedback is welcome!

NL(Normal Language){

it is almost complete.

this artificial language is called NL(Normal Language).

advertisement{

you can use NL fluently in just one month

It is the simplest, most efficient, easiest and most logical future global language.

NL dont have exception.

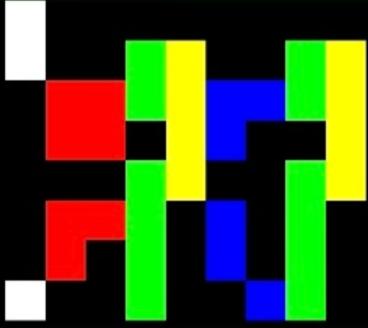
over 65% of words is compound word or antonym or arranged in an orderly way.

so you can remember words easily.

NL is phonetic script.

Since NL is a binary language, it can easily be converted into NL versions of barcodes, QR codes, and Morse code.

NL can express all meanings using only binary.

	$\lim_{x \rightarrow 0} \sin x = 0$
efficient alphabet	
latin alphabet	gdedpgihgdbcpgih
NL QR	

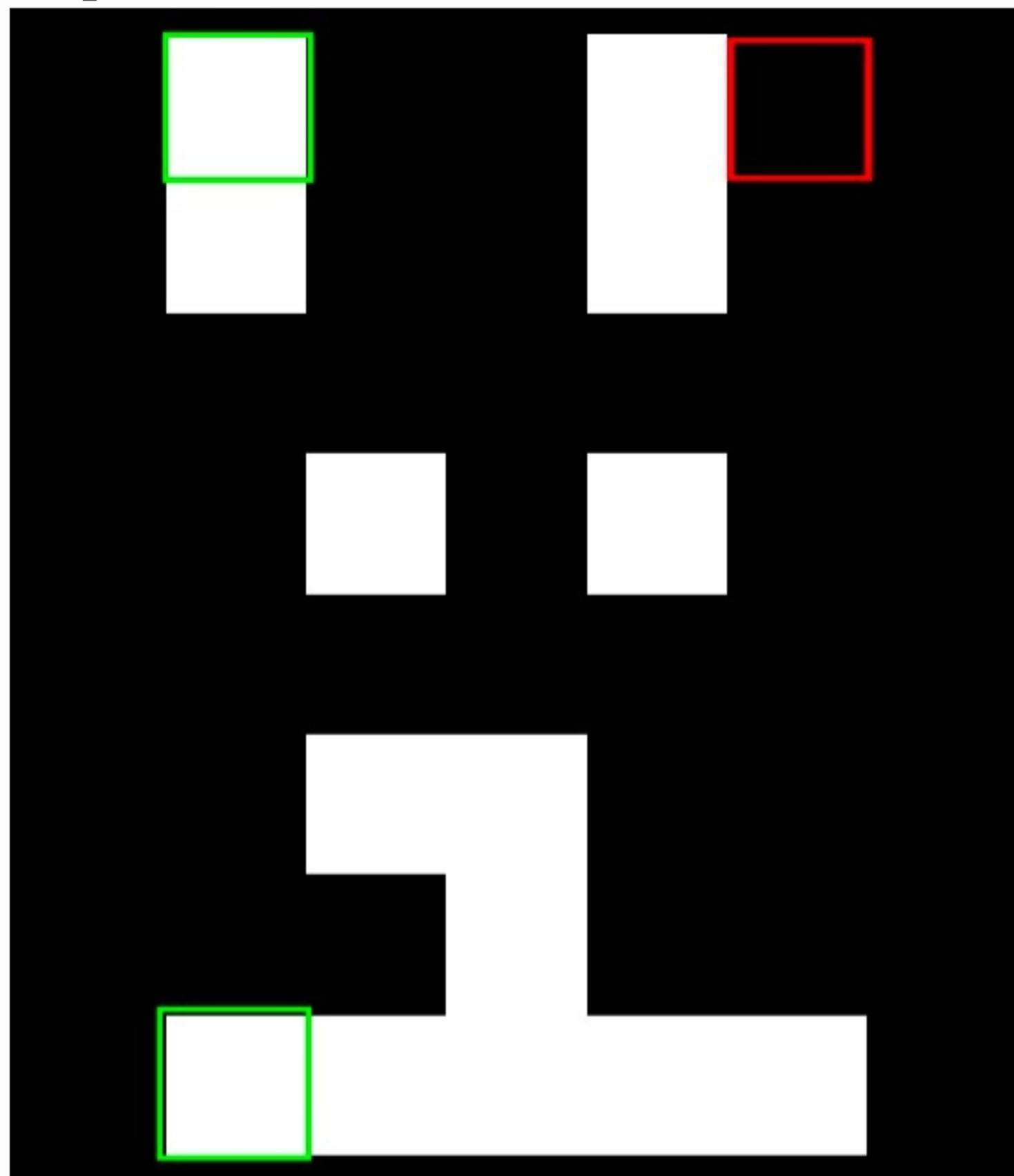
NL version QR code uses over 5 times less space than a QR code.
(the both image mean 'i am people')



NL version barcode is over 1.5 times more efficient than a barcode.(both image mean same)



NL code means the same from any angle like QR code.



vertex that should be black(0).

vertex that should be white(1)

one sentence has only one meaning.

ex 'A and B of C' can be interpreted as '(A and B) of C' and 'A and (B of C)' in english.

but in NL, 'af C bf A B' mean only '(A and B) of C'.

'bf A af C B' mean only 'A and (B of C)'.

'i know you' in NL is 'nidgbd'.

english is 80bit(in ascii code).

NL is 24 bit(one letter is 4bit).

NL is three times more efficient than English.

so NL reduces environmental pollution.

NL can be expressed in one, two, or three and more dimensions(pdf user can see image).



NL will become the global language.

}advertisement

alphabet{

alphabet is only 0 and 1 then it can be expressed as 16 latin letter.

a:0000, b:1000, c:0100, d:1100,

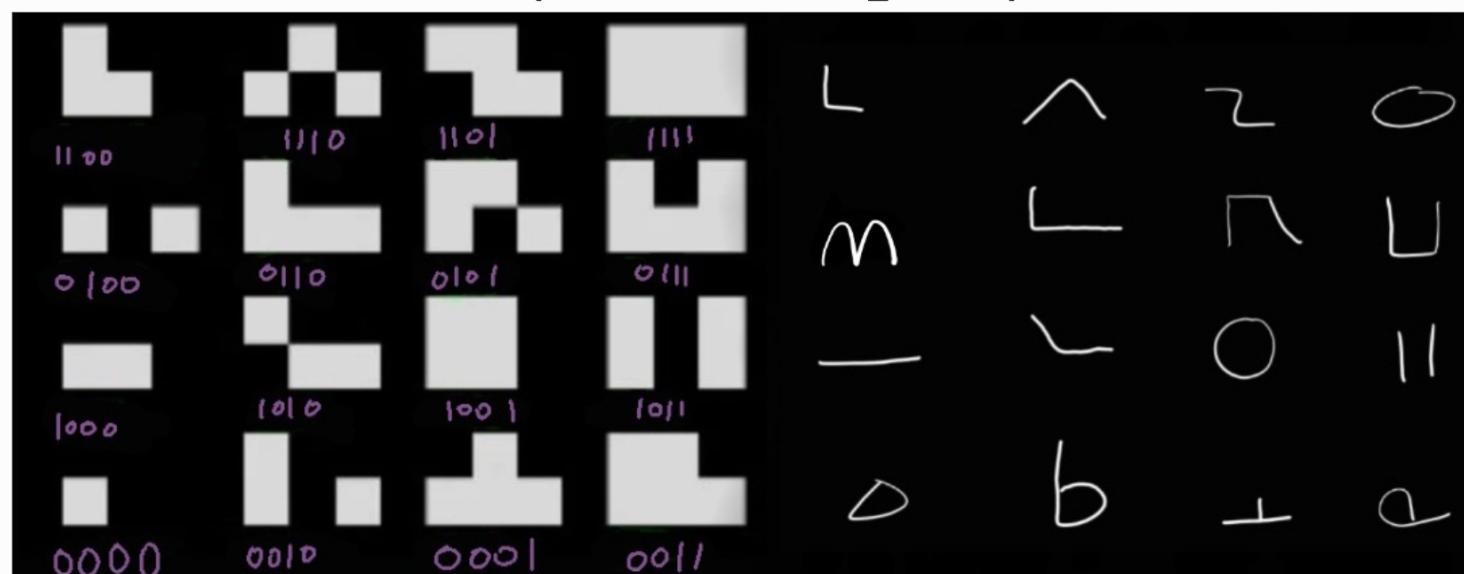
e:0010, f:1010, g:0110, h:1110,

i:0001, j:1001, k:0101, l:1101,

m:0011, n:1011, o:0111, p:1111.

NL uses only lowercase letters.

more efficient alphabet is here(only pdf user can see.
bottom to top writting alphabet):



8n-th bit determines length of word.

a word whose 8n-th bit is 1 make the word consist of more bit.(ex sentence '{0000101(1)0110000(0)} {0111000(0)}' consist of two words.

between of '(' and ')' is 8n-th bit.

between of '{' and '}' is one word.)

it means that word whose 2n-th letter is between of 'a'(0000) and 'h'(1110) consist of smaller quantity bit.

the example converted to latin letter is 'anga oa'.

so it dont need whitespace cuz the 8n-th bit plays the same role as whitespace in English.

but ill use whitespace to make you easily understand.

pronunciation{

The $(2n-1)$ th letter is pronounced as initial consonant.

and the $2n$ -th letter is pronounced as vowel also final consonant.

ex 'fc ee anba' is pronounced as 'neo da maeb'.

$(2n-1)$ th letter 'f' is pronounced as 'n'.

$2n$ -th letter 'c' is pronounced as 'eo'.

$(2n-1)$ th letter 'e' is pronounced as 'd'.

2n-th letter 'e' is pronounced as 'a'.

then 'anba' is pronounced as 'maembod' then it can be pronounced as 'maeb'.

pronunciation chart{

a:'moo'. b:'bo'. c:'leo'. d:'geu'.

e:'da'. f:'nae'. g:'kki'. h:'kwo'.

i:'ppoom'. j:'ttom'. k:'teom'. l:'peum'.

m:'sam'. n:'thaem'. o:'jim'. p:'jjwae'.

IPA version{

a:'mu'. b:'bo'. c:'rʌ'. d:'gʊ'.

e:'da'. f:'nɛ'. g:'kɪ'. h:'kʰwʌ'.

i:'pum'. j:'tom'. k:'thʌm'. l:'phʊm'.

m:'sam'. n:'sem'. o:'təm'. p:'təwə'.

}IPA version

}pronunciation chart

}pronunciation

}alphabet

EML(Efficient Memory Language){

lecture{

orientation{

NL is on left.

}orientation

basic{

grammar of NL is same as math.

transitive verb is function.

object is parameter.

structure of NL sentence is list of two objects.

NL sentence 'A B' mean 'B is A'.

NL dont have be verb.

NL use VOS word order.

sentence 'djca aoad' mean 'ore is solid'.

'djca' is 'solid'.

'aoad' is 'ore'.

some word must have parameter like function.

'mmfd oa oa' mean 'i love me',
'thing that love me is me'.

'mmfd' is function that must have parameter.

'mmfd' mean 'thing that love A'.

change capital letter into parameter value.

'oa' mean 'i' so 'mmfd oa' mean 'thing that love me'.

if subject is omitted, subject is 'i' so sentence 'mmfd oa' mean 'mmfd oa oa'.

this grammar can express every meaning.

ex 'The happy boy gives her a small book in the morning and smiles.' is expressed as '*ab gb if ab dmkg pf bida mlmb kjkc ac ajfg ajea ab dmdg boob*'.

}basic

<How to interpret a word dictionary>{

sometime few words in this language is changed to become perfect, some word in lecture can be old versions.

word dictionary is more accurate.

'fojb r!=dead thing.'{

word is expressed as it in word dictionary.

far left is NL word.

far right is meaning of this word.

between of NL word and '!' is word type.

'r' is an abbreviation for 'real'.

'r' type word mean word that can exist in reality 'r'.

}

'kjgg r!=alive thing.//anto die'{

//' is for an additional explanation.

'anto' is an abbreviation for 'antonym'.

'anto A' mean 'this word is antonym of A'.

The antonym form is the inversion

of the original word's bits.

0 becomes 1. 1 becomes 0.

inversion of

'fojb'(10100111,10011000) is

'kjgg'(01011001,01100110).

}

'oa r!=i.'//com express, this, symbol
that omit parameter.'{

'com' is an abbreviation for
'compound word'.

'com A,B' mean 'this word is
compound word of A and B'.

'oa' is compound word of 'kb' and
'ha' and 'ad'.

length of all that words is 24bit.

but i want to express word 'i' as
8bit word cuz word 'i' is used a lot.

The floor of 8 divided by 3 is 2.

2bit is from 'kb'(01,011000).

2bit is from 'ha'(11,100000).

4bit is from 'ad'(0000,1100).

result is 'oa'(01,11,0000).

}

'anga r!=people //pron de mensch'{

'pron' is an abbreviation for

'pronunciation'.

'pron de mensch' means that the pronunciation of this word is derived from the German word 'mensch'.

'de' mean german in ISO code alpha2.

}

'eb v!o=thing except A.'

between of '!' and '=' is parameter that this function must have.

'o' is an abbreviation for 'object'.

any object can be in object type parameter.

capital letter in far right is parameter value.

ex 'eb fojb' mean 'thing except dead thing'.

function in parameter have parameter.

}

'ha plo=this A.'

'p' is abbreviation for 'parameter'.

parameter value determines type of this function.

ex 'ha ih' is number cuz 'ih' is number.

}

'ih n!=0'{

'n' is abbreviation for 'number'.

}

'pc si!=continue speaking.'

'si' is abbreviation for 'sign'.

'si' type word dont interface all words.

ex sentence 'eb pc anga oa' mean
'eb anga oa' cuz word dont
interface 'si' type word.

You can continue what you were saying before by using word 'pc'.

}

'ad si!=symbol that omit parameter'{

some 'si' type have

}

'fc rlo,o=thing that ask B about A.'{

'!A,B' mean this function need 'A' and 'B' as parameter.

A is first parameter of NL word, the second one as B, and then C,

D, E, F, and so on.

parameter that this function must have is **object** and **object**.

}

'bg n!int=integer'{

'int' is an abbreviation for
'integer'.

'int' is for expressing only integer.

so it dont make parameter.

'bg' express integer number.

ex 'bg he' is 35.

first bit determines whether the

number is positive or negative.

if first bit is zero, number is negative.

integer 'he' is '111000010'.

(2nd bit)*1=1.

(3rd bit)*2=2.

(4th bit)*4=0.

(5th bit)*8=0.

(6th bit)*16=0.

(7th bit)*32=32.

$1+2+32=35$.

so it is 35.

}

'ka 2p|**o**-**o**-**n**=thing that become B
using A.'{

'2p' is '2rd parameter'.

type of word in 2rd parameter
value determines type of this
word.

ex type of 'ka anga kjgg' is 'r' cuz
'kjgg' is 'r'.

The symbol '-' means that if the
parameter is omitted by symbol
that omit parameter, the **next**
parameter is omitted too.

ad si!=symbol that omit one

parameter.

}

'gb p!s-n=reason why A2 is A1'{

's' is an abbreviation for
'sentence'.

's' is 'o-o'.

so 'A1' is first 'o'.

'A2' is second 'o'.

}

'bf plc o=A1 and A2 and....'{

'c o' mean 'this word have object type parameter until it met symbol that omit parameter'.

'A1' is first parameter.

'A2' is second parameter.

ex 'bf oa oa oa ad' mean 'me and me and me and me'.

}

}<How to interpret a word dictionary>

conversation example{

<basic 2>{

'mmfd bd oa', 'me fd bd oa'='thing
that love you' is me', 'i love you'{

mmfd rlo,n=thing that love A with B
percent intensity,//com
desire,normal,

me r!s,n=thing that hope A2 is A1
true with B percent intensity.

fd r!=normal,

'me fd bd' mean 'thing that hope
you are normal'.

}

<how to ask>{

'fc ne bd'='thing that ask you
about location of you is me',
'where are you?, i ask you'{

ne v!o=location of A.

}

'fc gb dnfd oa bd', 'thing that ask
you about reason why thing that
hate me is you is me', 'why do you
hate me?'{

fc r!o,o=thing that ask B about A.

gb pls,n=reason why A2 is A1.

dnfd r!o,n=thing that hate A://com
dread,normal,

if second parameter of 'fc' is
omitted, it is 'you' or 'i'(alone
think).

}

'fc jc dnea noba bd'='thing that
ask you about what happens if
thing that give up being worker is
you is me', 'what if you give up
being worker?'{

jc pls,n=what happens if A2 is A1.

dena r!o=thing that gave up being
A.

noba r!=worker.

}

'fc bb hmea oa bd'='thing that ask you about whether thing that see me is you is me', 'are you looking at me?'{

bb r!s=whether A is true.

hmea r!o=thing that see A://com sense, photon, .

}

'fc ec gjkb ha anga boba'='how do we save this person?', 'thing that

ask you about how we become
thing that save this people is me'{

ec r!s-n=how A2 become A1.

gjkb rlo,n=thing that save A.//com
reason,alive,..

ha plo=this A.

boba r!=we.//com and,i,you,o,..

}

'fc ee he id boba'='when we sense
each other?', 'thing that ask you
about time when thing that sense
each other is we is me'{

ee t!s=time when A is true.//com
what,thing that be B when A,

id r!=each other //com
what,interface,self,.

}

}<how to ask>

time{

'jb gjfg anga oa', 'jb gb fojb anga
oa' = 'thing that was thing that kill
people' is me', 'i killed people'{

jb plo=thing that was A.

gjfg own=thing that kill A //com
reason,die,.

eida o!=that.

gb pls,n=reason of A.

fojb r!=die.

'gb fojb anga' means 'reason why people is die'.

}

'**gc gifg eida** oa'='thing that was thing that kill that is me', 'i will kill that'{

gc plo=thing that will be A./ant
was A.

}

'eb jb fojb oa'='i have never died'{

}

}time

<symbol that omit parameter>{

'fc ad bd'='thing that ask you
about something is me', 'i ask you'{

ad si!=symbol that omit one
parameter.

bc si!=symbol that omit two

parameter.

cc si!=symbol that omit three parameter.

dc si!=symbol that omit four parameter.

omitted parameter become mean 'something'.

}

'dnfd bf bd dnfd bd bc oa'='i hate you and thing that hate you'{

dnfd r!o,n=thing that hate A.

bf plc w=A1 and A2 and....

in this situation

it should omit two parameter so
'bc' is used instead of 'ad'.

the first omitted parameter is 2th
parameter of 'dnfd'.

second is for finish 'bf'.

}

}<symbol that omit parameter>

<number>{

'ag'~'ph' is number word.

'bg da', 'lh'='1'{

'bg da' is '10000110,11000000'.

'lh' is '11011110'.

and see between of '(' and ')' in next.

'bg da' is '10000110,(110)00000'.

'lh' is '(110)1110'.

'ih'~'ph' is 'eni'(efficient number integer).

}

'ja kh fojb'='two dead thing'{

in original, 'kh' is '-1' but the quantity cannot be negative.

so in situation that dont need minus, it become positive integer.

so 'kh' become 2.

in original, 'ih'(0,00,11110) is '0'.

'kh'(0,10,11110) is '-1',

'lh'('1,10,11110') is '1'...

'ph'(1,11,11110) is 3.

in situation that dont need minus,
'ih' is '0',

'jh' is '1',

'kh' is '2'...

'ph' is '7'.

}

'jafojb'='quantity of dead thing'{

}

'dg fa jh'='1.25'{

dg n!int-n=real number.

'fa jh'='10100000.10011110'.

'fa' is in decimal area.

'jh' is 4 but in situation that dont need minus, number become positive.

}

how to command{

'be alive!'=>'fa kjgg'{

fa r!o,o=thing that orders B to
become A.

kjgg r!=alive.//ant die

}

}how to command

<how to use 'ea'>{

'mmfd bd oa', 'bd ea mmfd oa'='i
love you'{

first sentence mean 'i love you'.

second sentencee mean 'what i
love is you'.

'ea' is similar to 'what' of english.

}

'fc ad bd', 'bd ea kh fc'='i ask you'{

To know the meaning, you need to
list the parameters of all words in
sentence of paramrter of the 'ea'.

ea vlc n,s:'ea omit targeted
word'=what.

'kh' is not in sentence paramter of
'ea'.

'ea kh fc' mean 'second parameter of 'fc' ' cuz 'kh' is 2 in situation that dont need minus.

}

'nidg bd oa', 'bd ea nidg pa oa'='i know you'{

first sentence mean 'i know you'.

second sentence mean 'what i know is you'.

}

//i review document up to this point using 20250604version nl.

'cpmb ea jh mmfd bd', 'cpmb ea

'mmfd bd', 'cpmb ea mmfd pa
bd'='thing that dont know what you
love' is me', 'i dont know what you
love'{

explanation:

parameter of 'mmfd' is 'w,n'.

parameter of 'bd'=none.

if 'c n' parameter of 'ea' is
omitted, it automatically target
first parameter(in this case, it is
parameter 'w' of 'mmfd'(what it
loves)).

how to express using parameter 'n'
of 'ea'.

second sentence use 'n' using 'enl'.

that situation dont need minus so

'jh' become 2.

'jh' target first parameter of 'mmcd'.

how to express using word 'pa'.

Put the word 'pa' into content you want to express.

ex 'ea de pa bd'='what you sense'.
parameter of 'de' is 'what oneself
senses'.

}

}<how to use 'ea'>

<number and mathematical

expression>{

'mmfd cd ph'='i love you with
100percent intensity'{

}

'lh bd nh ad ph'='1+2=3'{

lh n!=1.

nh n!=2.

ph n!=3.

number word in sentence become
have parameter 'int'.

it is operator.

yeah, 'bd' is 'you' in original but it
is in math expression.

'bd' is 'plus'.

'ad' is 'equal'.

}

'ja v!n,**o**=A B.'

example of 'number' + 'word' in english is 'one apple'.

ex 'ja ih anga' mean 'zero people'.

ih=0.

If different type word like 'r' is in place of an 'n' type parameter, then 'n' is automatically omitted.

if parameter 'n' of 'ja' is omitted,

function 'ja' return quantity.

ex 'ja anga' mean 'quantity of people'.

you can omit number type parameter to express quantity of number using symbol that omit parameter.

ex 'ja ad ih' mean 'quantity of 0'.

}

}<number and mathematical expression>

//u

<logic and how to make new word>{

}<logic and how to make new word>

<advanced>{

}<advanced>

)conversation example

quiz{

answer{

}answer

}quiz

)lecture

word dictionary{

word{

aa si!=finish-all,

ba,

ca

da si!=enter.

ea plc ni,s:'ea omit targeted
word'=ohat./certainty

fa rlo,o=thing that orders B to
become A.

ga plo=only A.

ha p!o=this A.

ia p!o=value marker.

ja p!n,o=A B,

ka p!o-o-n=thing that become B
using A.

n=symbol-that-change-language,//
MTCL.

ni:proton,ni:neutron,ni:electron=ato
m

la

ma p!o=all,

na r!=fact,//com too and,A
include,real,true,o,.

o-n=into or,

oa=rli,//com express,this,omit word,

pa

=finish_standard,

ab r!o-o=A also B,//com be, and,.

int,n,para=symbol_that_change_parameter_to_word_type,

bb s=whether A is true //zero is true.cuz true is important.exist made false.one is exist.

c o=too and.

cb

db v!int=word as int A from onself.

int=name,

eb plo=thing other than A.

t!s=date,//com what,2ni,

fb

gb pls-n=reason of A,

o-o= B according to A,

hb plo,n=thing that can be A with
B percent probability./bisi
becoming.

ib

jb plo=thing that oas A./cant com
b when a,past,o,.

kb plo=thing that express A,

lb

mb

plo=average of A.

nb

ob

pb

silni=backspace,

ac plo-o=B that A include //com
also, thing that A include, .

bc sil=symbol that omit two
parameter.

cc sil=symbol that omit three
parameter.

dc sil=symbol that omit four

parameter.

ec r!<o-o>-n=how B become A.//
what,should,

n,{n,n,n}...=molecule,

fc r!o,o=thing that ask B about A.//
com command,express,

gc p!o=thing that will be A.//anto
A of past.,

o,n=say,

hc

!o=category-of,

ic p!o-o=intersection of A and B.//
com also,A include,A include,

!int=elementary-particle,

jc pls-n=result of that // auto
reason.

kc

lc !int-int-para...=too function.

mc=1plt-o=thing that be B when A.

!n,o=B that has id A,

nc r!o=thing that call A // com
command,sense,i,.

oc p!o=thing that becoming A.

!n,o=A th B // minus is from last.

pc si!=continue-say,

ad si!=symbol that omit one
parameter.

bd r!=you.//anto i,

cd p!<o-o>-n=thing that become A
using oneself also B.//
use,also,self,.

dd c=self,//no cw but intend com
word-as-id,+2id.

ed 2p!o-o=thing that be B with A.//
cooperation,,no cw but intend com
also,around,.

fd r!=normal.

gd int=A th function.

r!o=thing that make A.//com
reason,real,

plo=thing that has the
characteristic A.//characteristic

is for expressing '성질'

percent,

hd plo=word A.

id r!=each other // com
what, interface, self, .

jd

kd

ld c int=nl ipa.

md plo,n=thing that intend be A.

nd t!=now.

n,int...=phonetic.

od sil!int=version marker.

pd si!=cancel-express,

ae t!o=thing that be A.

be p!<c o>-n=or{.

ce p!o=other A.//com also,not,oac,.

de r!o=thing that sense A.

n=word as int,

ee t!s=time when A is true.//com
what,thing that be B when A,

fe r!o,o,t=thing that prohibit B
from being A.//com command,not,

o=text,

ge p!n,o=thing that similar to A.

he by plo=element of A,//include promise.

ie p!<c n,s>=what_with_category//com also,value marker_what,.

=name_or_var,

plo=essence of A.

je t!s=past of A,

c o->=array,

ke r!n=abnormal//anto normal,

!o-o->=converted,

le plo-o=between of A and B//com

me r!s,n=thing that desire A is true with B percent intensity.

!e n,o=cast,

ne n or r!o=location of A.

!str=argb,

oe

plo=thing that become A.

plo=thing that tends to become A.

pe o!int=word_as_situation_and_id.

af 2plo-o=B of A.//com
also,element-of,.

int=operator,

bf plc o=and{//com or,most

silc int,n=duplicate_word,

cf r!s=that A2 is A1.

df r!s,n=thing that dread A with B percent intensity,//anto desire,

ef 2p!o-<o,n>=B that is by A.// also,result,fact,.

ff

gf t!s=future of A,//anto past of A.

hf

if r!o=thing that have A.//anto element,.including promise

jf r!s=thing that know A is true.// com know,sen,true,whether,.

kf r!o,o=thing that ansoers B

saying A.//anto ask,

r!s=directly cause, //com
reason, over, nh,

If o=around of A.

!o-o=most-near,

mf si!=word as context.

o=minus of A.

p!o-n=part of A.

nf

of c int=foreign language.

pf

ag

bg n!int:-1 in minus=integers,

cg n!int=value,

dg n!o-o=and A B.

eg n!int-n:int=real number,

fg n!o-o=or A B.

gg n!o-o=xor A B.

hg n!o-o=nor A B.

ig

n!=other-math,

jg

kg n!o=less than A, //A>

int-int...=string,

lg n!o=bellow than A, //A>=

mg n or t!o-o=A plus B.

n!o=numerical.

ng n!o-o=A times B.

n!int=unknown,

og n!o-o=A square B.

pg n!=x.

n!=number.

ah

bh n!o-o=A root B.

ch n!o-o=A divided by B.//anto
square.

dh n!o-o=A minus B.//anto plus.

eh n!o=exceed than A.//anto
bellow,,A<=

fh n!o=more than A.//anto less,,A<

gh

hh n!=y.

ih n!=0.

jh n!=4.

kh n!=-1.

lh n!=1,

mh n!=2,

nh n!=2,

oh n!=3,

ph n!=3,

etc{

glea p!n=n-th word,

bmba r!=world.

cnad=situation //com sen,0,.

ekke r!=logic //how,true,situation,.

cjba r!=transistor,

bjda r!=god, //pron ru bog

ojda r!=crops,

bjdg r!=true. //com
whether, zero, zero,

njba r!=real.

kjba=empire,

ljba r!=shop,

r!o=thing that famous in A,

akba r!=app,

bkba r!=weapon,

ckba r!=game,

dkba r!=ideology,

fkba r!=scholarship,

gkba r!n=job,

mkba r!=city,

nkba

elph r!o=thing that focus be A.//
com intend,ph,.

dlba r!o=map of A,

elba r_int=digit,

flba r_s:purpose-s,n=thing that
efficient than B for be A,

eida o!=that.

bmba r!o=south of A.

cmba r!o=west of A.

dmba r!o=north of A.

emba r!o=east of A.

hmba r!=home.

kmba r!o=think that thank A.

lm̄ba r!o=thing that revenge A.

nmba r!o=entropy,

inba p!o=A dollar,

!o=thing that never be A.//com
not,all,A of past.

r!o=As.//com hoo many,over,one,.

onba r!=mistake,

doba p!o=information of A,

joba r!=cpu,

akga r!=money,

noba r!=worker,

ooba r!o,n=thing that enjoy be A.

poba p!n or s=wages.

apga r!n=company.

epba r!o=thing that vote A.

apba=time.

bpba t!o=A second,

cpba t!o=À minute,

dpba t!o=À hour,

epba t!o=À day,

pmib p!o=what À be.//about À,

alhg p!o=thing that be a.//anto À
be.

gpba t!o=À week,

hpba t!o=À month,

ipba t!o=À year,

ppba =write is until here,

aica r!n=mosfet,

dica p!o=thing that start be À.

eica plo,n=middle of A.

fica plo=thing that finish be A.

o,n=thing that gives birth to A,//
com launch,baby,.

kica r!=baby,

ajca r!n=political party,

bjca r!=gas,

cjca r!=liquid,

djca r!=solid,

ijca r!=book,

mjca rlo,o,n=thing that write A at B,

rlo=thing that see A.//com

sense, light,

n:'number of id'-id:'words'-
str:'structure of
parameter'=expressing-mean-of-
word,

dkca r!o=daily of A,

ekca r!o=life of A,

r!=food,//com
what,inhale,also,normal,animal,.

kjkc r!=thing that laughs.//
com,express,fun,

hkca r!o=structure of A,

ikca r!o=principle of A,

kkca r!o,n=thing that think A is fun,

mkca r!=gun,

lkkb rlo=thing that inhale A.

dlca rlo=thing that aim A.

jlca n=var, //of js

klca n=let, //of js

olca rlo=terrain of A.

cpmb rlo=thing that dont know A, .//
anto know, .

fmca r!=video.

gmca rlo=thing that select A.

kmca plo,o=similarities.

pmca pln=thing that split into A.

r!s,n=thing that help A,//com
intend,reason,

n=level of language.

elgf r!o,n=thing that feel angry by
A,//com intend,ruin,.

knca o=attribute.

Inca,

mnca r!=planet.

nnca r!=sun.

onca r!o=standard of A.

coca o:target,o:location=pros,

doca o,o=cons,

eoca ni=natural-phenomenon,

hida

gida plo=limit of A,

eknb s=evidence of A.//
what,need,know,whether,.

jida rlo=thing that looking for A.

nida plo-o=between of A and B,

aoad rlo=ore.

plo=age of A.

cjda =novel.

djda =legend.

gjda rlo=thing that sex with A.

hjda pregnancy.

ijda =birth.

kjda ni=money.

bkda o=value of A.

ekda=capital.

hkda n or o=rgb.

cnnda n=id...=IPA.

fnnda o=speed of A,

gnnda o=momentum,

hnnda o=rotation,

inda o=angular-momentum,

knda o=electric-charge,

mnda ni=plant,

clba o-o=thing that promise B to A.

aoda o,o:standard=inconvenient,

coda o=symbol of A.

eoda =religion.

goda =civilization,

joda o,n=thing that forgive A,

moda n=pressed,

poda o,o:standard=important,

niea o=thing that eat A,

oiea o=thing that cheo A,

piea o=thing that gulp doon A,

ajea o=thing that digest A,

=excrete,//com launch,shit,.

cjea =shit,

gjea o,n=thing that repeat A.

jjea o,o,n=thing that sorry.

ongg o=thing that go in A//com
go,inside,.

gmob o=inside of A.

fjda o,n=thing that have purpose
of being A,

jke a o:is,o,t=difficulty,

mkob o,n=thing that ignore A,

clea s=request,

ejba=objective.//
what,beilive,true,logic,.

flea o=thing that trade with A,

nlea ni=peninsula,

amea id=QR code.

fnfg animal.//pron en animal.

emga o,n=thing that rule A,

alba o=law of A,

kmea id, str...=UTF8,

dnea o,n=thing that was given up
on being A.

=initial-consonant,

gneā=oil,

jnea o=thing that subscribe A.

hifa=document,

kifa o=thing that put on A,

nifa o:standard,o,n=rank,

pifa id:location-of-standard-
mark=prevent-error,

njfa=mountain,

jkfa o=thing that looking for A,

glfa o=plan,

plo=property of A.

bifa o=meter,

cifa o=metet^2,

difa o=meter^3,

goic s,s,s,c s=for-loop, // 'for(int
i=0;i==3;i+=2)' = 'goicjhadocagphocagl
hadocag'

hmga=thing that feel emotion.//
pron jp kanjou.

n-o=max,

s=but.//com reverse,sen

id:<number of oord>-ni:<targeting

parameter>-id<coord>-
id<structure>=mean of function,

o-o:of=distinguish,//

o=A as well,

mmkd o-o=difference between A
and B.

=justice.//objectice,equality..

mjdd o=peace.

boob o,n=thing that feel pleasure
by A.

gmgd o:who-o:theme=win,

gmda=sound,

gmbg=music,//com sound,pleasure,

fneg=shift,//anto motionless,

mlcd o=out,

nmbg o,o,n=thing that dont need A
to B.

gmnb o=thing that think that
enemy is A,

jmob o,n=thing that feel beauty,

fojb =die,

cpjc o=thing that A include,

bida=newton,//pron
neoton=strength,

mkfa o=thing that for A,//pron en
for,

Imoe o,n=thing that become B
using A also oneself.//com
use,sen,self,.

giga =country,//pron kr گ,

anba=people//pron de mensch,

ilnc=heart,//pron en hearts,

letter//pron kr ل,

enfg o=thing that launch A,//anto
inhale,.

nigf o=thing that include A,//anto
thing that A include,.

dome o=thing that fight with A,//
anto peace,

objg o,n=thing that feel pain by

A,//anto pleasure,

fnoe o,n=think that should not be
A,//com dont need,sen,self

kjgg n=alive,//anto die,

epjg o,n=no,//anto yes,

jlbg o,n=thing that think that A is
ugly,//anto feel-ugly

clob o-<o,n>=thing that need A to
be B,

jkdg o=thing that think A is
friend,//anto enemy,

dmne o=in,//anto out,

jlje o:who-o:theme=lose,//anto win,

gjfg o,n=thing that kill A,

coea o=name of A./com
what,wac,express,sen,target_what,

ajda o=A of past./com also,oas,o,

apba o=A of future./com
also,will,o,

=phonetic-alphabet,/com
express,sound,alphabet,

who-is,/com ask,be,.

pp=white-doarf,/com ohite,star,
=refuce,/com

o=thing that get A./com
become,have,.

ajea plo=that A./com also,that,

o=thing that impact A,//com
reason,shift,

kklb =motionless.

nidg o=thing that know A,//com
include,information,

=delete,//com
reason,become,not,real,

ajab c o=c also,//com also,also,

akac c o=c-of,//com of,of,

c o=c-by,//com by,by,.16,

mnmf c o=c-intersecion,

ekfe=pointed//
what,indicate,omit-2-gramar,

o,n=thing that study A.//com
becoming,know.

nicf s=thing that know situation
A.//com knoo,sen,.

gjfd o,n=thing that normalize A.//
com reason,normal,.

o,n=break,

nice o-o-n=thing should not be A
to be B.//com not-need,also,self,

cjoe o-o-n=thing that should be A
to be B.//need,also,self,

bmdg o=thing that greets A.//pron
de ade,.

o,o=distance,//meter,from-until,

ajec o=pointed A,//com
also,pointed,

ajea o=that A,//com also,that,

=right-of,//com ohat,use,normal,

=gay,//hope,breed,gender,self,

ekkb =language,//com
what,need,express,o2,

fmdf o=bye,//com
command,go,not,i,

ekfd=nl,//com normal,language,

o=type,//diffrence,one,

aiba=elementary particle.

biba=electron neutrino,

ciba=muon neutrino,

diba=tau neutrino,

eiba=photon,

fiba=gluon,

giba=electron,

hiba=up quark,

iiba=down quark,

jiba=strange quark,

kiba=muon,

liba=charm quark,

miba=tau,

niba=bottom quark,

oiba=o boson,

piba=z boson,

ajba=higgs boson,

bjba=top quark,

kjgc o-o=thing that give A to B.//
com reason,have,

o-o=thing that teach A to B.//com
reason,know,

o,o=thing that shows B to A.//com
reason,sense,

o,n=kind,

o=symbol that convert A into and type.

dndfd o,n=thing that hate A,//com
dread,normal,

mmfd o,n=thing that love A with
100percent intensity //com
hope,normal,

=side,

amkg o,o=weak, //com energy,less,

amfh o,o=strong, //com energy,more,

bmkg o-o=short, //com meter,less,

bmfh o-o=long, //com meter,more,

cmkg o-o=narrow,//com
meter2,less,

cmfh o-o=wide,//com meter2,more,

dmkg o-o=small,//com meter3,less,

dmfh o-o=big,//com meter3,more,

angf o=front of A.

bngf o=front left of A.

cngf o=left-of,

dngf o=behind-left-of,

engf o=befind-of,

fngf o=behind-right-of,

gngf o=right-of,

hngf o=front-right-of,

changed.//com become,not,A
be,self,.

o=human-race,

kked r!o=thing that denies A.//com
express,sen,not,true,.

o=understand,

o,n=thing that think that think A is

cute, //com

think, sentence, parameter, less,

s,m=thing that calculate A, //
intend,knoo,result

o:into,o,n=move, //com go, from-
until,

filb o,n=thing that dont curious
about A, //com oish,self,knoo,

kld o,n=think that wonder curious
about A, //com oish,self,knoo,

o,n=thing that want to sex with
A, //com,

mkia o:into,o,n=change, //com
reason,become,

bmoe o,n=something that feels
comfortable be A. //com

happy, sen, self, .

one o,n=thing that feel
uncomfortable be A, // anto
comfortable.

gjkb r!o-n=thing that heal A, // com
reason, alive,

ekde o=features of A, // com
difference, average, self,

mlmb r!=women. // bsf pussy.

dmdg r!=men. // anto oomen.

o,n=thing that want to be A, // com
hope, self,

enter, space=see-back,

boba=we, // com and, i, you, o,

o,n=yes,//com

goba o=thing that think A,

gnhb

str!=code.

cjmd o=gender,//com of,animal

omit,whole,
o=middle,

o=water-per,//com
intersection,oater,

o,n=thing that listen A.///com
sense,sound,

o=earth-oar,//com fight,earth,

o=treaty-organization,

```
 }etc
```

```
 }word
```

```
 id{
```

```
 cast{
```

```
 ag=word,
```

```
 bg=r-w,
```

```
 cg=n-w,
```

```
 dg=t-w,
```

```
 }cast
```

```
 english{
```

```
 b=space,
```

```
 }english
```

```
 body{
```

ag=brain,

bg=head,

cg

dg=hand,

eg=eye,

fg=mouth,

gg=face,

hg=genitalia,

ig=wing,

jk=loin,

mg=nappe,

}body

not{

af=difference-set,

bf=

time,

cf=purpose,

}not

function{

official is starting with 0.

da=differential.

fa n-n:from-n:until=integral.

ha=buffer gate.

ja=not.

la n:base-n=log.

na n:'from'-<n:'until'-n>=right-hand-lim.

pa n:'from'-<n:'until'-n>=left-hand-lim.

ed n-<n-w>=lim A->B.

```
eb s:'bottom'-n:'top'=sigma,
```

```
bc n=sin A.
```

```
bb n=cos A.
```

```
bg n=tan A.
```

```
i=factorization,
```

```
}
```

```
other-math{
```

```
ba=lowest,
```

```
ca=most,
```

```
}other-math
```

```
constant{
```

```
da=-most,
```

```
fa=most,  
ha=-infinity,  
ja=infinity,  
la=natural,  
na=imaginary,  
pa=pi,  
bb=speed-of-light,//photon
```

```
}
```

```
language{  
af=ML,  
bf=CL,  
}
```

```
}id
```

```
}word dictionary
```

```
}EML(Efficient Memory Language)
```

official account{

youtube: [https://youtube.com/
@normallanguage?
si=4E60EKgSuZa90JAj](https://youtube.com/@normallanguage?si=4E60EKgSuZa90JAj)

reddit:

email(feedback etc):
g wahagsinsa@gmail.com

}official account

rule{

This language is not complete and
is constantly changing, so you
must indicate the version when
using it like 'nl60125'(6m1d25y)
month->day->year.

if you should use integer 0, but 0
is expressed as 'aa' then trash bit
is appear. then 'ba' is 0.

license{

license can be changed.

this document and language have
copyright.

you can use this language as free.

}license

number{

'x1000010'

if x is 0,minus.

if x ia 1,plus.

before bit is small.
it is efficiency.

10000000=+0.

11000000=+1.

01000000=-1.

}number

reading way in NL QR{

in 1 dimension, '10' is at first for expressing reading direction.

in other dimension, it is
'10000011'.

or

'1011000000000011'

read like ➔⬇

rows should not 8,

}reading way in NL QR

for developer{

The efficiency of inaccurate expressions and accurate expressions should be similar to same.

pronunciation chart order is low herz->high herz.

exact expression is important than efficiency.

one sentence must mean only one mean.

purpose of language is only communication and efficient.

0 cannot be used for counting order.

exception should none.

relative things must have standard.

sentence 'people i' means 'i have over 70% of features of people'. it is called 'ratio word'.

10. arrangement order of word is 'number->most common->'.

word formation rule{

word that need parameter{

1: meaning of word whose parameter is all is what other word can express.

2: relative word.

transitive verb.

test sentence{

input word your wonder to @.

Forgotten @ repeats itself.(ex
history word dont need parameter.
try input history of. is it owe?)

people in @ love you.

There should be no discrimination
based on @ (that is features of
discrimination word. so 'gender of'
word legal)

}test sentence

}word that need parameter

antonym{

bit of antonym is inverse of original word so 0 become 1, 1 become 0.

word that should come first{

priority-est is

1.low number:can(0%)->cant,

2.original:desire()->desire(not).

3.many quantity:women is manier than men.

}word that should come first

if specific word have lot of antonym, created by antonym mechanism is 'root(no slave,no king)-

>number(0% can=cant)-
>purpose(weapon,treatment
tool,,intend, evolution)->difference
set(dont know,know').

}antonym

compound word{

1.'know' is 'have'+'information-of'.

2.'girlfriend'='girl'+'love' cuz
original is 'wae'+'girl'+'love'+'omit
gramr'.it should keep word order.

3.'teethbrush'='teeth'+'brush'.

4.phonetic loanword:'russia'-
>'romsimna(russia in russian. silent
consonant become 'n')'.
'higgs particle'->'(kims('h'->'k'))'.

if new compound word overlap with

other compound word, element
word of new compound word
become small.(ex '00,10' become
'01'. then if area of element is too
small, use XOR).

if length of bit of element of
compound word is not power of
two, element of compound word
such as symbol that omit grammar,
ea, of, and, or, also is omitted to
be power of two.

small bit word is first in situation
whose order doesn't matter.

}compound word

}word formation rule

}for developer

}rule

other{

if NL have problem NL is changed
never mind other.

Creator Verification

Code:'20@@@0226@017128.

pn:@@05139@@@8 in 2024/11/18.
JSM. SE. GIGADESADE. GOHW. SEMI'.

2025y6m8d updated.

prohibits anything similar to this.

thanks!{

in 2025y6m4d, dero told me about
polish notation so i decide to use
it for math expression in
2025y6m6d.

}thanks!

}other

history{

i thinked if we use weird language,
we become abnormal.

i liked science. i read science. but
i angry cuz english is ambiguous.
so i make it.

}history

research data{

using NL ipa.

/'pa/ similar to /bbua/ than /
pupua/.

is it lowest herz?{

my think{

sin{

m:100.b:140.d:160.n:180.'p:300.'t:500.
'k:700.'c:7000.'j:12000.t:16000.

}sin

tri{

d:130hz.l:190hz.

}tri

squ{

c:11500.

}squ

saw{

d:30hz.|:40hz.

}saw

d|mbn(p?)Pp(j?)(h?)TtKk(Ch?)sJjSst

d|m(q?)bn(j?)(p?)
(k?)'p(x?)'t'k(z?)c'j'ct

uyoraeiw'u'w'y'i'o'r'a'e

}my think

others think{

f0

mnvbdqlcjxzptk'p't'k'c'j

mnlbqdcxzptk'k't'p'j'c

uyoraeiw'u'w'y'i'o'r'a'e

}others think

}is it lowest herz?

}research data

element of language{

WL is world language.

EL is for efficient.

CL is Cpu Language.

ML is Memory Language.

}element of language

useless from here

useless from here

useless from here

useless from here

ECL{

exam{

//js to ecl

var a=0.
0,o,variable,1.

}exam

qeqzejix'as,

word{

a=finish command,

b id=to other language,

c ,

d=line break,

e ,

f ,

g=finish standard,

h=finish sentence,

i

j

k

l

m

n
o
p
ab,
bb,
cb,
db,
eb,
fb,
gb =external memory to cache
memory,
hb ML ss,ML s,ML s,CL s=for-loop,
ib
jb,
kb id,w=access to external
devices,
lb
mb
nb
ob
pb
ad=omit-gramar,
bd=omit-2-gramar,

cd=omit-3-gramar,
dd n:+20 in e=omit-n-gramar,

}word

}ECL,

WL{

it is incomplete.plz ignore WL,go
to BWTL,
sound of WL{

xxyyyy'yy'{

xx is frequency type,

yy is frequency,

}

(2bit:s or t or q or a.6bit:32n+32= of 8bit,

pronunciation expressed using NL ipa.

sin

32n{

aa:du. ea: bu. ia:'tu. ma:ju.

ab:lu.eb:'t'i.ib:dud.mb:dul.

ac:bws.ec:mus.ic:dup.mc:cub.

ad:y.ed:do.id:bul.md:qul.

ae:rm.ee:ym.ie:'trm.me:'twm.

}32n

maybe 128n{

aeim

ae:'vu.ee:mo.ie:'cu.me:dum,

af:'vo.ef:mom.if:'tum.mf:'co,

ag:om.eg:'pon.ig:'tus.mg:'nom,

ah:'pos.eh:'cus.ih:'trm.mh:'kuk,

ai:'tym.ei:'tu.ii:'pun.mi:'po,

aj:'jo.ej:'tos.ij:'k.mj:'pum,

ak:'tom.ek:'mo.ik:mu.mk:mr,

al:m'r.el:'t'r.il:'tw.ml:'tr,

I . . 'tun I . . 'tom I . . 'ty I . . 'ku ,

I . . 'p'r I . . 'toq I . . 'tob I . . 'prm ,

I . . 'kos I . . 'kr I . . 'j I . . 'jum ,

I . . 'kw I . . 'tul I . . 'cin I . . 'tin ,

I . . 'pis I . . 'tis I . . 'cun I . . 'ti ,

I . . 'pin I . . 'p'is I . . 'tim I . . 't'i ,

I . . 'k'is I . . 't'is I . . 'p'ipp'p'in ,

}maybe 128n

}sound of WL,

WCL{

}WCL

WML{

<Number of expression
objects,Number of
cases,frame,Minimum force,what
compare>
-><dimension,bit,strength,frame,cas
e>

sign{

rank of priority{

sev is several,
r is relative,
infi is infinity,
}rank of priority

}sign

a=finish-sentence-and-standard,
ad=finish-sentence,
world<0,0,0,0>,
none<0,0,0,0>,

time<0,0,0,2>,
elementary-particle<1,1,sev,1,19>,
exist<1,1,r,1,infi>,

atom<1,100~300,1,1.67×10^-27kg>,
molecule<1,300~,1,2×1.67×10^-27kg>,
appear/disappear<1,2,2,1>,
none change/change(s,r,p=<1,2,2,1>
light/heavy,(become light/become
heavy=<1,2,2,2>,
(become low/become
many=<1,2,2,mc^2>,
square<1,1,1,srpr>,
number<1,2,1,srpr>(1,
inner/out<1,2,1,1,7>,
this i/this u<1,2,1,2,1>,
under/down/left/up/right/
on<1,6,1,1,7>,
line<2,1,1,2,6>,
move to (under/down/left/up/
right/on=<1,6,2,2,14>,
stay/move<1,2,1,2,14>,
split/combine<(1'2.,2,2,2×mason,13>,

decrease/gain<(1'2.,2,2,3,13>,

}WML,

}WL,

}NL(Normal Language)

