

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 : <https://youtube.com/@normallanguage?si=4E60EKgSuZa9OJAj>

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 50% 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.

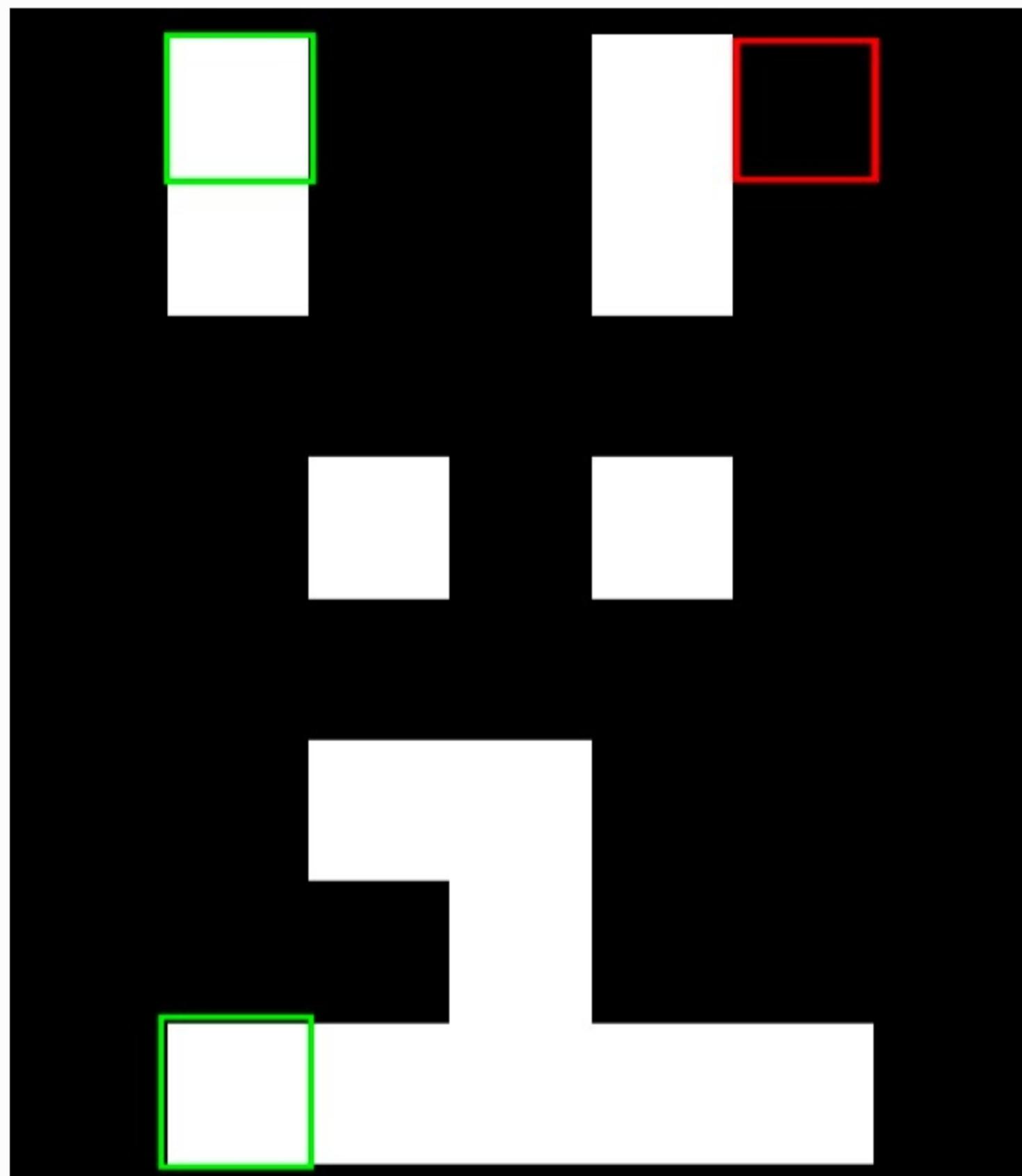
NL version QR code uses over 5 times less space than a QR code.



NL version barcode is over 1.5 times more efficient than a barcode.



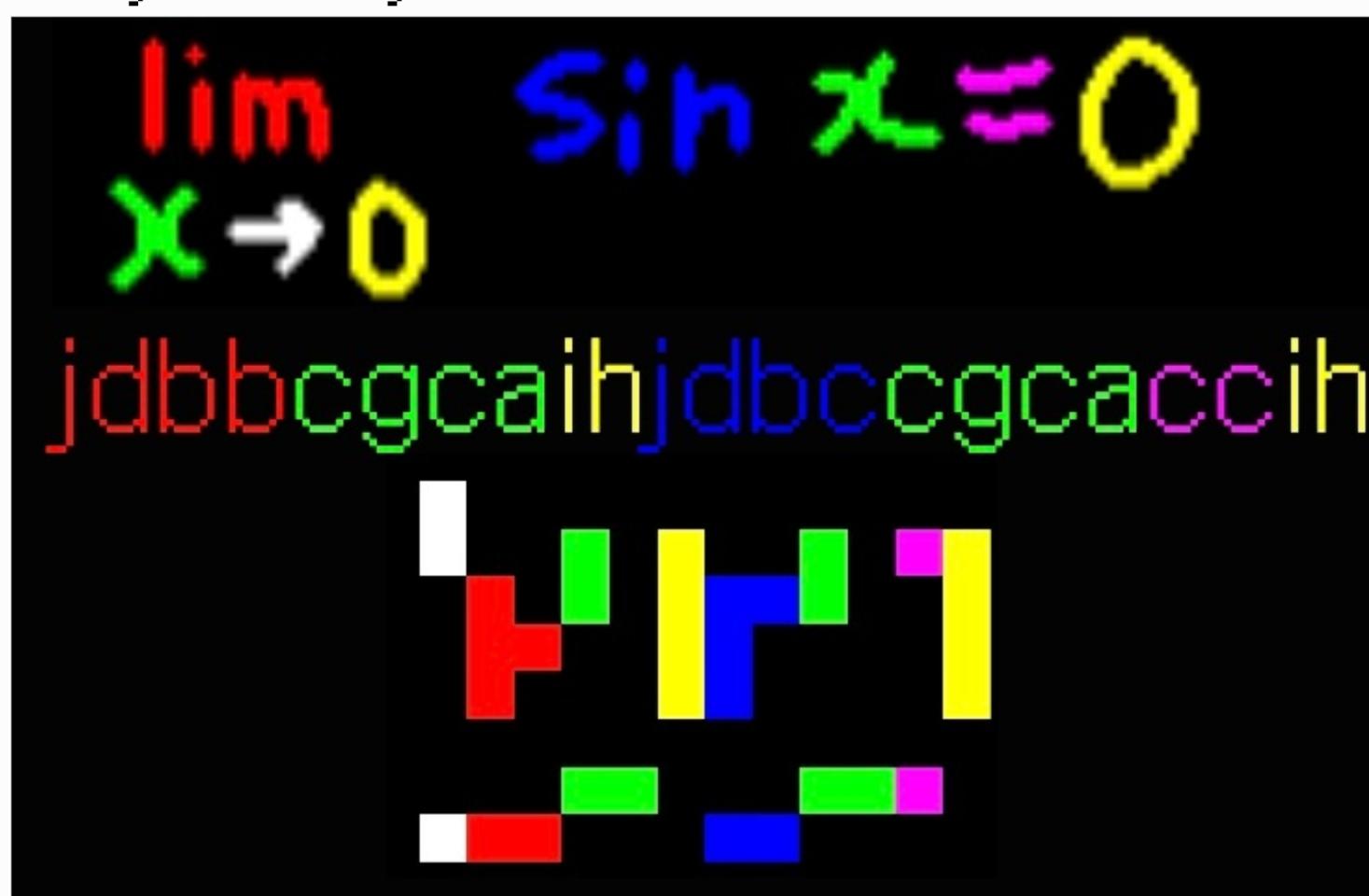
it means the same from any angle like QR code.



vertex that should be black(0).

vertex that should be white(1)

it can express all meanings using only binary.



one sentence has only one meaning.

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

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

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

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.

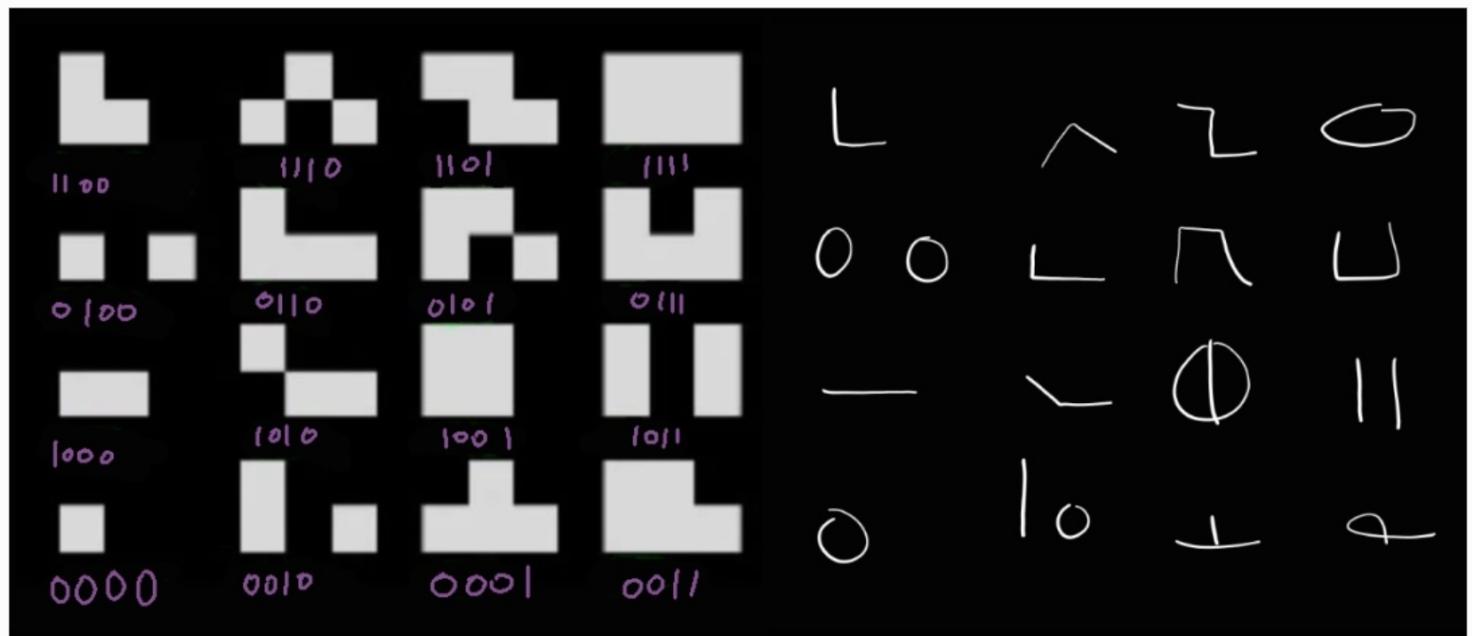
i:0001. j:1001. k:0101. l:1101.

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):



in binary, The $8n$ -th bit 1 makes the word have more bits.

if $8n$ -th bit is 0, it has separator.
(ex sentence '(00001011,01100000,) (01110000)' is consist of two words)

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

it means $2n$ -th letter between 'a'(0000) and 'h'(1110) is letter that have separator.

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

but ill use whitespace to make you
easily understand.

pronunciation{

'fc ee lanba' is pronounced as 'neo
da maeb'.

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

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

$(2n-1)$ 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 'maembo' but 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:'gw'.

e:'da'. f:'nɛ'. g:'kɪ'. h:'kʰwʌ'.

i:'pum'. j:'tɔm'. k:'tʰʌm'. l:'pʰwum'.

m:'sam'. n:'sɛm'. o:'tɛm'. p:'tɛwɛ'.

}IPA version

}pronunciation chart

}pronunciation

}alphabet

EL(Efficient Language){

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.

sentence 'anga oa' mean 'i am people'.

NL dont have be verb.

'he bd, oa' mean 'thing that sense
you is me'.

word 'he' is function.

so it must have parameter.

'bd' is in parameter of 'he'.

'he bd' mean 'thing that sense you'
cuz 'bd' mean 'you'.

verb and object always attach to
each other.

'oa' mean 'I'.

this grammar can express
everything.

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>{

'**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'.

word 'r' is thing that can exist in reality.

dead thing can exist in reality so this word type is 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,o,'{

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

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

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.

}

'he r!w=thing that detect A.'{

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

'w' is an abbreviation for 'word'.

any word can be in word type parameter.

capital letter in far right is parameters.

A is first parameter of NL word,
the second one as B, and then C,
D, E, F, and so on.

ex 'he fojb' mean 'thing that
detect dead thing'.

word in parameter can have
parameter.

ex 'he he fojb' is 'thing that
detect thing that detect dead
thing'.

}

'ha plw=this A.'{

'p' is abbreviation for 'parameter'.

The type of this word is determined by parameter of this word.

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'.

ex sentence 'he pc oa oa' is same as 'he oa oa' cuz 'si' type word dont impact grammar.

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

}

'ja v!n,w=A B.'{

symbol ';' mean 'and'.

this word must have **number** and **word** type parameter.

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

ex 'ja ih fojb' mean 'zero dead

thing'.

ih=0.

if parameter 'n' of 'ja' is omitted,
function 'ja' return quantity.

ex 'ja fojb' mean quantity of dead
thing.

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

but someone want to express
quantity of number.

you can omit 'n' type parameter
using symbol that omit parameter.

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

}

'ka 2p!w-w-n=thing that become B
using A.'

'2p' is 2rd parameter.

type of word in 2rd parameter
determinate type of this word.

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.

ex sentence 'ka ad eida' mean
'thing that become something'

using something is that' and 'that is user'.

eida w!=that.

}

'bg n!int=integer'{

'int' is an abbreviation for
'integer'.

'int' is for expressing only integer.

so binary in 'int' dont make parameter.

ex 'bg he' is 33.

first bit determines whether the number is positive or negative.

if first bit is zero, number is negative.

integer 'he' is '11100010'.

(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.

}

'gb pls-n=reason why A2 is A1'{

's' is an abbreviation for
'sentence'.

's' is '**w-w'** except math expression.

so 'A1' is **first 'w'**.

'A2' is **second 'w'**.

}

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

'c w' mean 'parameter of this word
is between this word and symbol
that omit parameter. then

parameter type is 'word'.

ex 'bf,oaoaoa,ad' mean 'me and me and me'

}

}<How to interpret a word dictionary>

example{

<basic>{

'*angā* *oa*'='I am *people*'{

explanation:

it is expressed as

'(0000101,1,0010000,0,)(0111000,0,)'
in binary.(8n-th bit is important)

NL is on the left.

anga r!=people //pron de mensch.

oa r!=i.

mean of NL sentence 'word word'
is '2nd-word is 1st-word'.

NL dont have be verb.

}

'he bd oa=' 'thing that sense you' is
me', 'i sense you'{

he r!w=thing that sense A.

bd r!=you.

}

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

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

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

fd r!=normal,

'oa' is in parameter 'n' of 'mmfd'.

if diffrent type word is in
parameter 'n', parameter 'n' is

automatically omitted.

}

'**gjfg eida** jb oa', '**gb fojb eida** jb oa' = 'thing that kill that' is me of past'; 'i killed that'{

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

eida w!=that.

jb p!w=A of past.

gb p!s,n=reason of A.

fojb r!=die.

'gb fojb that' means 'reason it die'.

}

'*gjfg eida* gc oa', '*gjfg eida* gc'='thing that kill' that is me of future', 'i will kill that'{

gc p!w=A of future,//anto A of past.

if parameter of 'gc' is omitted, it is 'me of future'.

}

}<basic>

<how to ask>{

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

ne v!w=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!w,w=thing that ask B about A.

gb pls,n=reason why A2 is A1.

dnfd r!w,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!w=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!w=thing that see A.//com
sense,photon,.

}

'fc ec gjkb ha anga boba'='thing that ask you about how we become thing that save this people is me', 'how do we save this person?'{

ec r!w-w-n=how B become A.

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

ha p!w=this A.

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

}

}<how to ask>

<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'{

dnbd r!w,n=thing that hate A.

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

bd r!=you // anto i.

bc si!=symbol that omit parameter.

oa r!=i // com, express, this, o..

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

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

second is for finish 'bf'.

}

}<symbol that omit parameter>

<number>{

'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.

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

'ag'~'ph' is number word.

}

'jafojb'='quantity of dead thing'{

}

'dg fa jh'='1.25'{

dg int-n=real number,

'fa jh'='10100000.1001110'.

'fa' is in decimal area.

'jh' is 4 but in situation that dont need minus, all 'e n' become positive.

}

how to command{

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

fa w,w-t=thing that orders B to become A.

kjgg=alive./die

}

}how to command

<how to use 'ea'>{

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

explanation:

ea c n,s:'ea omit targeted word'=what,

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

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

parameter of 'bd'=none.

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

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

second sentence use 'n' using 'eni'.

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>{

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

}

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

lh=1.

kh=2.

ph=3.

number in sentence become have parameter 'int'. it is operator.

yeah, 'bd' is 'you' in original but it is in parameter 'int' of number.

'bd' is 'plus'.

'ad' is 'equal'.

}

}<number and mathematical expression>

//u

<logic and how to make new word>{

}<logic and how to make new word>

<advanced>{

}<advanced>

<symbol that express 'parameter
and property' of dictionary>{

}<symbol that express 'parameter
and property' of dictionary>

}example

quiz{

answer{

}answer

}quiz

}lecture

word dictionary{

word{

aa si!=finish-all,

ba,

ca

da si!=enter.

ea p!c ni,s:'ea omit targeted
word'=what.

fa r!w,w,t=thing that command.

ga p!w=only A.

ha p!w=this A.

ia

ja pln,w=A B,

ka plw-w-n=thing that become B
using A.

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

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

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

ma plw=all,

na

w-n=into or,

oa=r!i,//com express,this,omit word,

pa

=finish_standard,

ab r!w-w=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 w=too and.

cb

w=minus,

db

int=name,

eb p!w=thing other than A.

p!n,w=thing that similar to A.

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

fb

gb p!s-n=reason of A,

w-w= B according to A,

hb

ib

jb p!w=A of past,//cant
com,also,past,o,

kb p!w=thing that express A,

lb r!=true;//com

whether,one,o,one,o2.

mb

nb

ob r!=real,

pb

w!=thing,//com include in,all,,.

si!ni=backspace,

ac p!w-w=B that A include //com
also,thing that A include,.

t!s=when,

bc si!=symbol that omit two
parameter.

cc si!=symbol that omit three

parameter.

dc si!=symbol that omit four parameter.

ec r!<w-w>-n=how 1A2 become 1A1.// what,should,

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

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

gc p!w=A of future,//anto A of past.,

w,n=say,

hc by p!w=element of A,//include promise.

!w=category-of,

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

!int=elementary-particle,

jc p!s-n=result of that.//anto
reason.

kc

lc

mc

!n,w=B that has id A,

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

oc p!w=thing that becoming A.

!n,w=Ath B.//minus is from last.

pc si!=continue-say,

ad si!=symbol that omit one parameter.

bd r!=you://anto i,

cd p!w=other A//com also,not,wac,.

dd c=self,//com word-as-id,+2id.

ed 2p!w-<w,n>=B by A//B that A include,result,fact,.

fd r!=normal.

gd

p!w=thing that has the characteristic A//characteristic is for expressing '성질'

percent,

hd p!w=word A.

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

jd

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

ld

md p!w,n=thing that intend be A.

nd

n,int...=phonetic.

od s!int=version marker.

pd si!=cancel-express,

ae p!w-<c
n,s>=what_with_category,//
also,what,.

be plc w-n=or{,

ce

de int=A th function.

n=word as int,

ee

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

w=text,

ge

he r!w=thing that sense A.

ie

=name_or_var,

p!w=essence of A.

je t!s=past of A,

c w->=array,

ke r!n=abnormal // anto normal,

!w-w->=converted,

le p!w-w=between of A and B // com

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

!e n,w=cast,

ne n or r!w=location of A.

=word-result,

!str=argb,

oe p!w,n=thing that can be A with
B percent probability,

p!w=thing that become A.

p!w=thing that tends to become A.

pe w!int=word_as_situation_and_id.

af 2p!w-w=A of B.//com
also,element-of,.

int=operator,

bf plc w=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

ff

!int-int-para...=two_function,

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

hf

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

jf r!s=thing that know A is true//

com know, sen, true, whether, .

!w=average,

kf r!w:thing, w:to=answers B saying
A//anto ask,

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

If

!w-w=most-near,

mf

p!w-n=part of A.

nf

w=now,

of

pf si!=word as context.

ag

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

cg n_int=value,

dg

eg n_int-n:int=real number,

fg n_w=efficieny-bracket,

gg

hg

ig

n_other_math,

jg

kg n_w=less than A, //A>

int-int...=string,

lg n_w=bellow than A, //A>=

mg n_w=numerical, //it is only
number.

ng n_int=unknown,

og

pg n_=number,

ah

bh

ch,

dh

eh n_w=exceed than A // anto
below,,A<=

fh n_w=more than A // anto less,,A<

gh

hh

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,

iiiba r_=logic,

cjba r_=transistor,

bjda r_=god,//pron ru bog

ejba r_=crops,

kjba=empire,

ljba r_=shop,

njba r_w=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_w=thing that focus be A.//
com intend,ph,.

dlba r_w=map of A,

elba r_int=digit,

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

eida r_=that.

bmba r_w=south of A.

cmba r_w=west of A.

dmba r_w=north of A.

emba r_w=east of A.

hmba r_=home.

km̄ba r_w=think that thank A.

lm̄ba r_w=thing that revenge A.

nm̄ba r_w=entropy,

in̄ba p_w=A dollar,

r_w=As // com how many, over, one, .

on̄ba r_=mistake,

doba p_w=information of A,

joba r_=cpu,

akga r_=money,

noba r_=worker,

ooba r_w,n=thing that enjoy be A.

poba p_n or s=wages.

apga r_n=company.

epba r_w=thing that vote A.

apba=time.

bpba t!w=A second,

cpba t!w=A minute,

dpba t!w=A hour,

epba t!w=A day,

pidg t!w=what A be.//about A,

fpmb t!w=thing that be A.//anto
what A be.,

gpba t!w=A week,

hpba t!w=A month,

ipba t!w=A year,

ppba =write is until here,

aica r!n=mosfet,

dica p!w=thing that start be A.

eica p!w,n=middle of A.

fica p!w=thing that finish be A.

w,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 r_w,w,n=thing that write A at B,

r_w=thing that see A,//com
sense,light,

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

dkca r_w=daily of A,

ekca r_w=life of A,

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

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

hkca r!w=structure of A,

ikca r!w=principle of A,

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

mkca r_=gun,

lkkb r_w=thing that inhale A.

dlca r_w=thing that aim A.

jlca n=var,//of js

klca n=let,//of js

olca r_w=terrain of A.

cpmb r_w=thing that dont know

A, // auto know, .

fmca r_=video,
gmca r_w=select,

kmca p_w,w=similarities.

pmca p_n=thing that split into A.

p_s,n=thing that help A, // com
intend, reason,

n=level of language.

elgf r_w,n=thing that feel angry by
A, // com intend, ruin, .

knca w=attribute.

Inca,

mnca r_=planet.

nnca r_=sun.

onca r_w=standard of A.

coca w:target,w:location=pros,

doca w,w=cons,

eoca ni=natural-phenomenon,

hida

gida p_w=limit of A,

jida r_w=thing that looking for A.

nida p_w-w=between of A and B,

aoad r_=ore,

p:'n:w,,w:n'w=age of A.

cjda =novel,

djda =legend,

gjda r_w=thing that sex with A.

hjda pregnancy,

ijda =birth,

kjda ni=money,

bkda w=value of A,

ekda=capital,

hkda n or w=rgb,

cnnda n-id...=IPA,

fnnda w=speed of A,

gnda w=momentum,

hnnda w=rotation,

inda w=angular-momentum,

knnda w=electric-charge,

mnda ni=plant,

clba w:'promise to be',w:'with
who'=promise B to A,

aoda w,w:standard=inconvenient,

coda w=symbol of A,

eoda =religion,

goda =civilization,

joda w,n=thing that forgive A,

moda n=pressed,

poda w,w:standard=important,

niea w=thing that eat A,

oiea w=thing that chew A,

piea w=thing that gulp down A,

ajea w=thing that digest A,

=excrete,//com launch,shit,.

cjea =shit,

gjea w,n=thing that repeat A,

jjea w,w,n=thing that sorry.

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

gmob w=inside of A.

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

jkea w:is,w,t=difficulty,

mkob w,n=thing that ignore A,

clea s=request,

flea w=thing that trade with A,

nlea ni=peninsula,

amea id=QR code,

emga w,n=thing that rule A,

alba w=law of A,

kmea id, str...=UTF8,

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

=initial-consonant,

gneā=oil,

jnea w=thing that subscribe A.

hifa=document,

kifa w=thing that put on A,

nifa w:standard,w,n=rank,

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

njfa=mountain,

jkfa w=thing that looking for A,

glfa w=plan,

w=property,

bifa w=meter,

cifa w=metet^2,

difa w=meter^3,

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

=thing that feel emotion,

n-w=max,

s=but.//com reverse, sen

id:<number of word>-ni:<targeting
parameter>-id<word>-

id<structure>=mean of function,

w-w:of=distinguish, //

w=A as well,

mmkd w-w=difference,

emkg=justice,

mjdd w=peace,

boob w,n=thing that feel pleasure
by A,

gmgd w:who-w:theme=win,

gmda=sound,

gmbg=music, //com sound,pleasure,

fneg=shift, //antostay,

mlcd w=out,

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

gmnb w=thing that think that
enemy is A,

jmob w,n=thing that feel beauty,

fojb =die,

cpjc w=thing that A include,

bida=newton,//pron
newton=strength,

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

lmoe w,n=thing that become B
using A also oneself./com

use, sen, self,.

eiea =country, //pron kr 𠂊,

anba=people//pron de mensch,

ilnc=heart, //pron en hearts,

letter//pron kr 𠂔,

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

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

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

objg w,n=thing that feel pain by
A, //anto pleasure,

fnoe w,n=think that should not be

A,//com dont need,sen,self

kjgg n=alive,//anto die,

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

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

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

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

dmne w=in,//anto out,

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

gjfg w,n=thing that kill A,

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

ajda w=A of past //com also,was,o,

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

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

who-is //com ask,of,real,

pp=white-dwarf //com white,star,

=refuce //com

w=thing that get A //com
become,have

ajca p_w=that A //com also,that,

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

kklb =stay,

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

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

ajab c w=c also,//com also,also,

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

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

mnmf c w=c-intersecion,

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

w,n=study,//com becoming,know,

nicf s=thing that know situation

A.//com know, sen.,

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

w,n=break,

nine w-w-n=thing should not be A
to be B.//com not-need,true
situation,self,

cjne w-w-n=thing that should be A
to be B, //need,true situation,self,

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

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

ajec w=pointed A.//com
intersection, pointed,

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

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

=gay,//wish,self,breed,gender,self,

ekkb ni=language,//com
what,use,express,o2,

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

ekfd=nl,//com normal,language,

w=type,//difference,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=w boson,

piba=z boson,

ajba=higgs boson,

bjba=top quark,

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

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

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

w,n=kind,

w=symbol that convert A into and

type.

dndfd w,n=thing that hate A,//com
not-wish,normal,

mmfd w,n=thing that love A,//com
wish,normal,

=side,

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

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

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

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

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

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

dmkg w=w=small,//com meter3,less,

dmfh w=w=big,//com meter3,more,

angf w=front of A.

bngf w=front left of A.

cngf w=left-of,

dngf w=behind-left-of,

engf w=befind-of,

fngf w=behind-right-of,

gngf w=right-of,

hngf w=front-right-of,

w=human-race,

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

w=understand,

w,n=thing that think that think A is
cute, //com
think, sentence, parameter, less,

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

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

filb w,n=thing that dont curious
about A,//com wish,self,know,

kld w,n=think that wondrr curious
about A,//com wish,self,know,

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

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

bmoe w,n=something that feels
comfortable be A,//com
happy,ser,sel,.

onoe w,n=thing that feel
uncomfortable be A,//ant
comfortable.

jjkb w,t,n=thing that heal A,//com
reason,live,

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

mlmb r!=women //hi pussy.

dmdg r!=men //anto women.

w,n=thing that want to be A,//com
wish,self,

enter,space=see-back,

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

w,n=yes //com

goba w=thing that think A,

gnhb

cjmd w=gender //com of,animal

omit,whole,
w=middle,

w=water-per,//com
intersection,water,

w,n=thing that listen A//com
sense,sound,

w=earth-war,//com fight,earth,

w=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,

jg=loin,

```
mg=nape,
```

```
}body
```

```
<nig>operator{
```

```
ib=and,
```

```
jb=or,
```

```
kb=XOR,
```

```
lb=NOR,
```

```
bd=plus,
```

```
cd=minus,
```

```
dd=times,
```

```
ed=divided-by,
```

```
fd=square,
```

```
gd=root,
```

```
}math
```

```
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,

db n:'from'-<n:'until'-s>=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

number{

'x1000010'

if x is 0,minus.

if x ia 1,plus.

before bit is small.

it is eficency.

10000000=+0.

11000000=+1.

01000000=-1.

}number

}word dictionary

}EML(Efficient Memory Language)

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,

j

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,

}EL(efficient Language),

Which word should come first?{

priority-est is

- 1.low number:can(0%)=cant,
- 2.original:desire()=desire(not).
- 3.many:women is manier than men,

}Which word should come first?

element of language{

WL is world language.

EL is for efficient.

CL is Cpu Language.

ML is Memory Language.

}element of language

word that need parameter{

1:meaning of word that dont has parameter is what other word can express.

2:relative word.

}word that need parameter

reading way{

in 1 dimension, '10' is at first for expressing reading direction.
in other dimension,it is
'10000011'.

or

'101100000000001'

read like → ↓

rows should not 8,

}reading way

length of word{

8n bit is one word.

if [8n]th bit is 0, this word have separator(it is spacing word).

if [8n]th bit is 1, long length word(it is called connecting word).

not 8n bit mean is [x]th word.

example{

00000000/*<-8n bit is 0 so short*/

0000000100000000/*8n bit is 1 so
become long*/

}example

}length of word

license{

1.this document and language have
copyright.

2.you can use this language as
free.

}license

official account{

youtube: [https://youtube.com/
@normallanguage?
si=4E60EKgSuZa9OJAj](https://youtube.com/@normallanguage?si=4E60EKgSuZa9OJAj)

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 'version 010525' day-

>month->year.

pronunciation chart order is low
herz->high herz.

exact expression is important than
efficiency.

if length of bit of element of
compound word is not power of
two, in first, element of compound
word such as symbol that omit
grammar is omitted it is power of
two.

in second, ea, of, and, or, also are
omitted to

one sentence mean only one mean.

purpose of language is only
communication and efficient.

in antonym, 0 become 1, 1 become 0 from original.

0 cannot be used for counting.

if specific word have lot of antonym, order of priority is 'interdependence(slave!=king)->direction(small!=big)->number(0% can!=cant)->purpose(weapon!=treatment tool)->difference set(know!=dont know)'.

exception should none.

relative things must have standard.

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).

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->'.

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

for developer{

word formation rule{

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')).

}word formation rule

}for developer

}rule

other{

1.if NL have problem NL is changed
never mind other.

2.maded by germany.

3.Creator Verification

Code:'20@@@0226@017128.

pn:@@05139@@@@8 in 2024/11/18.

JSM. SE. GIGADESADE. GOHW. SEMI'.

4.2025y5m31d updated,

5.

6. Prohibits anything similar to this.

7.

}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

WL{

useless from here

useless from here

useless from here

useless from here

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

xxyyyyyyy{

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

)WML,

)WL,

)NL(Normal Language)

