

I recommend that you view it as a pdf and dark mode(for font and image): <https://github.com/Fhres126/nl/blob/main/nl.pdf>

if that link dont work: <https://glki.netlify.app/nl.pdf>

youtube: <https://youtube.com/@normallanguage?si=4E6OEKgSuZa9OJAj>

our chat room: <https://discord.gg/VhHBjXbpkz>

all feedback is welcome!

norlang{

this artificial language is called norlang(Normal Language) in english.

advertisement{

norlang will become the international language.

norlang dont have exception.

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

so you can remember words easily.

one sentence has only one meaning.

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

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

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

'i know you' in norlang is 'nbbd'.(This is pronounced differently from English. it is easy to pronounce)

english is 80bit(in ascii code).

norlang is 16 bit(one letter is 4bit).

norlang is five times more efficient than English.

so norlang reduces environmental pollution.

norlang helps easy communication.

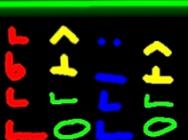
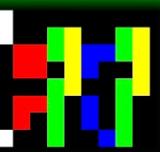
norlang brings peace and helps technology grow.

norlang is phonetic script.

norlang versions of barcodes and QR codes is called NL QR.

Since norlang is a binary language, it can easily be converted into NL QR and norlang version morse code.

norlang can express all meanings using only binary.

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

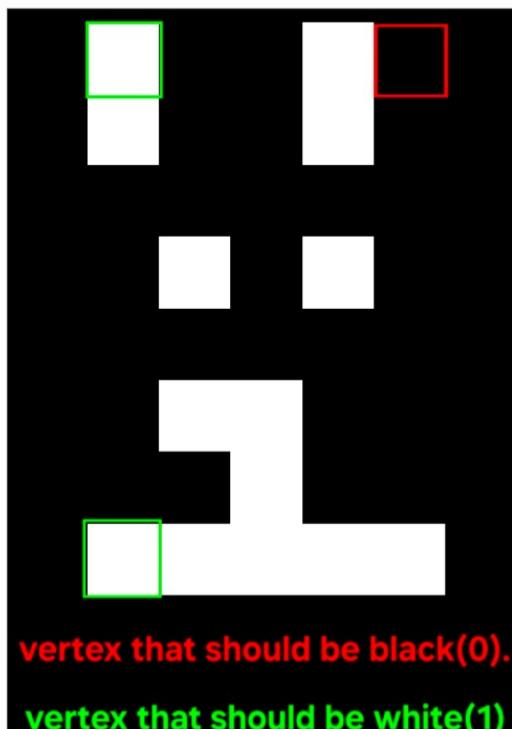
NL QR1 is over 1.5 times more efficient than a barcode.(both image mean same)



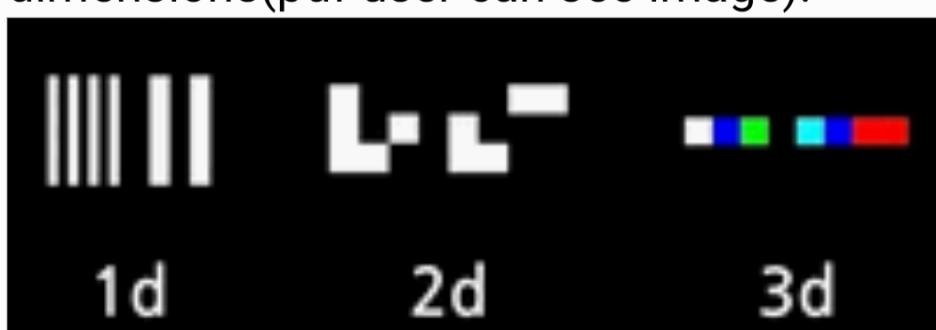
NL QR2 uses over 5 times less space than a QR code.(the both image mean 'i am people')



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



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



you can use norlang fluently in just one month

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

norlang can be used in programming.

it is called ECL.

}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.

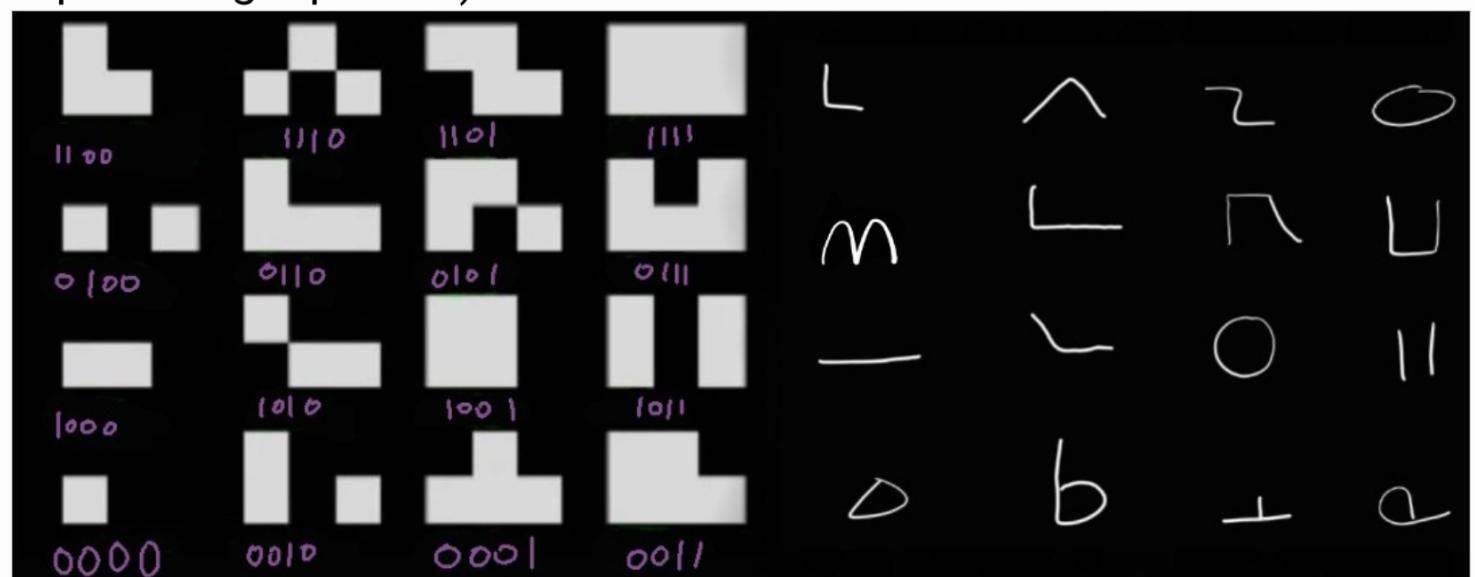
e:0010. f:1010. g:0110. h:1110.

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

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

norlang uses only lowercase letters.

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



\* 8n-th bit determines length of word.

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

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

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

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

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

so it deosnt need whitespace cuz the 8n-th bit 0 plays the same role as whitespace in English.

but ill use whitespace to make you easily understand.

pronunciation{

\* Two letters make one syllable.

\* (2n-1)th letter is pronounced as initial consonant.

\* 2n-th letter is pronounced as vowel also final consonant.

eg 'fc ee anba' is pronounced as 'neo da maemboo'.

(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'.('e' is 'da' in pronunciation chart)

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

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

pronunciation chart{

sound consists of decibels (dB) and hertz (Hz).

it is arranged with lower Hz sounds coming first.(eg m(100hz) < s(14000hz))

so it is not biased toward any specific culture.

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

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

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

m:'sam'. n:'jaem'. o:'ssim'. p:'jjwae'.

IPA version{

a:'mu'. b:'bo'. c:'rʌ'. d:'gw'.

e:'da'. f:'nɛ'. g:'ki'. h:'twʌ'.

i:'pum'. j:'kʰom'. k:'tʰʌm'. l:'pʰum'.

m:'sam'. n:'tœm'. o:'sim'. p:'tœwɛ'.

}IPA version

}pronunciation chart

}pronunciation

}alphabet

EML(Efficient Memory Language){

lecture{

basic{

- \* grammar of norlang is same as math.
- \* transitive verb is function.
- \* object is parameter.
- \* norlang dont use be verb.
- \* norlang use VOS word order.
- \* norlang sentence 'A B' mean 'B is A'.
- \* structure of norlang sentence is list of two objects.  
eg 'djca aoad' mean 'ore is solid'.

'djca' mean 'solid'.

'aoad' mean 'ore'.

\* some word must have parameter like function.

'jd bd oa' mean 'i think you', 'thing that think you is me'.

\* 'jd' is function that must have parameter so 'jd bd' is one object.

'bd' mean 'i' so 'jd bd' mean 'thing that think you'.

'oa' mean 'i'.

\* if subject is omitted, subject is 'i' so 'i think you' is expressed as sentence 'jd bd' and 'jd bd oa'.

this grammar can express every meaning.

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

To learn in detail, you need to know how to interpret a dictionary.

<How to interpret a word dictionary>{

'fojb r!=dead thing.'{

word is expressed as it in word dictionary.

\* far left is norlang word.

\* far right is meaning of this word.

\* between norlang word and '!' is word type.

\* 'r' is an abbreviation for 'reality'.

only thing that can exist in reality is reality type object.

}

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

'//' is for an additional explanation.

'ant' is an abbreviation for 'antonym'.

\* 'ant 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.

eg antonym of 'fojb'(10100111,10011000) is  
'kjgg'(01011001,01100110).

}

'jd r!o=thing that think A.'{

between '!' 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 input value of parameter of this function.

eg 'jd oa' mean 'thing that think me'.

}

'pc si!=continue speaking.'

'si' is abbreviation for 'sign'.

'si' type word dont interface all words.

eg sentence 'jd pc bd oa' mean 'jd bd oa' cuz word dont interface 'si' type word.

You can use 'pc' word to continue what you were saying before.

}

}<How to interpret a word dictionary>

any word can be in object type parameter.

'jd oa' is one object.

'jd jd oa' is one object and mean 'thing that think thing that think me'.

'jd ad oa'='i think', 'thing that think something is me'{

ad si!=symbol that omit one parameter.

paramrter of 'jd' is omitted by 'ad'.

}

'fmeg'='hello', 'thing that greets is me'{

fmeg r!o=thing that greets A./pron de ade,,

subject is omitted so subject is 'i'.

}

'higina coea oa'='my name is higina', 'higina is name of me'{

coea v!o=name of A.

}

}basic

intermediate{

<How to interpret a word dictionary>{

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

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

A is first input value of this function of norlang word, the second one as B, and then C, D, E, F, and so on.

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

eg 'fc aoad oa' mean 'thing that ask me(oa) about ore(aoad)'.

}

'ha p!o=this A'{

'p' is abbreviation for 'parameter'.

parameter value determines type of this function.

eg 'ha fojb' is reality type cuz fojb is reality type object.

}

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

'2p' mean '2rd parameter'.

type of 2rd input value determines type of this word.

eg type of 'ka oa kjgg' is reality type object cuz 'kjgg' is reality type object.

The symbol '-' means that if the **parameter** is omitted by symbol

that omit parameter, the next parameter is omitted too.

you can omit all parameter of 'ka' with only one 'ad'.

}

'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'.

'kb ha ad' mean 'thing that express this'.

total 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.

}

'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 p!c o=A1 and A2 and....'{

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

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

}

}<How to interpret a word dictionary>

<how to ask>{

'fc anga bd'='what is people? i ask you', 'thing that ask you about people is me'.{

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

}

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

ne v!o=location of A.

if second parameter of 'fc' is omitted, the second parameter mean 'you'.

}

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

gb p!s,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).

'gb dnfd oa bd' mean 'reason why you hate me'.

}

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

jc p!s,n=what happens if A2 is A1.

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

noba r!=worker.

}

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

ja r!s=whether A2 is A1.

dmea 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 r!o,n=thing that save A./com reason,alive,.

ha p!o=this A.

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

}

'fc ee de 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 A2 is A1./com what,thing that be B when A,

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

}

}<how to ask>

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

time{

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

jb p!o=thing that was A.

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

eida o!=the thing.

gb p!s,n=reason why A2 is A1.

fojb r!=die.

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

}

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

gc p!o=thing that will be A./ant was A.

}

'eb jb fojb oa'='i have never died', 'thing other than thing that was dead thing is me'{

eb v!o=thing other than A.

jb p!o=thing that was A.

}

}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 p!c o=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>

'ka af oa bode dmea bd'='i see you using my eye', 'thing that  
become thing that see you using eye of me'{

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

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

bode r!=eye./com body,see,.

dmea r!o=thing that see A./com sense,light,

}

'kjge fleg'='i speak fleg'{

gmkb r!o=thing that speak A./com sound,express.

fleg=norlang,/com normal,language,.

}

"='im from'{

}

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

mmfd r!o,n=thing that love A by B percent./com hope,normal,

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

fd r!=normal,

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

}

}intermediate

upper intermediate{

<How to interpret a word dictionary>{

'ih n!=0'{

'n' is abbreviation for 'number'.

}

}<How to interpret a word dictionary>

<how to use 'ea'>{

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

'bd ea mmfd oa' mean 'what i love is you'.

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

}

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

ea v!c n,s=what.//ea remove targeted parameter by A.

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

function in parameter sentence of 'ea' is 'fc'.

The parameters of 'fc' are 'what it asks about' and 'what it asks'.

'kh' is 2 in situation that dont need minus.

so 'ea' target second parameter of 'fc'.

}

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

first sentence mean 'i know you'.

second sentence mean 'what i know is you'.

The 'pa' is in parameter of 'nb'.

parameter of 'nb' is 'what it know'.

}

'ea mmfd bd'='ea jh mmfd bd'='ea mmfd pa bd'{

}

'ae anga mmfd oa'='people that i love', 'lovely people'{

ae p!o-<c n,s>=A that C2 C1./C2 is subject,,C1 is verb,,com  
also,what,

'ae anga mmfd oa' mean 'ab anga ea mmfd oa'.

ab p!o-o=A also B.

}

}<how to use 'ea'>

}upper intermediate

advanced{

<How to interpret a word dictionary>{

'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.

eg '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.

}

'if p!o=thing that have A./ant element'{

he p!o=element of A.

'if' is '0 0 0 1 1 0 1 0'.

'he' is '1 1 1 0 0 0 1 0'.

antonym of 'he' seems 'id'(00011100).

antonyms have an order.

An element must exist for the owner to exist.

so word 'element' consists of lower bits than 'have'.

then antonym of 'he' seems 'ih' but 'ag'~'ph' is for number.

word like 'element' is called 'former word'.

'gidd o-o-n=thing that should be A to be B,//need,also,self,'{

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

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

dd c=self.

The floor of 16 divided by 3 is 5.

But the length of element word in the compound word must be a power of 2.

}

}<How to interpret a word dictionary>

'ag'~'ph' is number word.

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

'bg da' is '10000110,11000000'.

'lh' is '11011110'.

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

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

'lh' is '(110)11110'.

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

}

'cb 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'.

}

'cbfojb'='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.

}

'cb v!n,no=A B.'{

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

eg 'cb 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 'cb' is omitted, function 'cb' return quantity.

eg 'cb anga' mean 'quantity of people'.

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

eg 'cb ad ih' mean 'quantity of 0'.

}

}<number>

<mathematical expression>{

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

}

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

}

}<mathematical expression>

}advanced

//i review document up to this point using 20250612version  
norlang.

<logic and how to make new word>{

}<logic and how to make new word>

NL QR{

convert alphabet to binary.

NL QR1{

draw '10'.

0 is black or white.

1 is opposite color of 0.

input it to next of '10'.

draw '11' at last to finish.

read from '10' to '11'.

eg 'oa'->'01110000' -> '10,01110000' -> '10,01110000,11'.

if last is not finished by '01' then you can omit '11' of last.

eg '10,01110000'.

}NL QR1

NL QR2{

draw '10000011'

draw bit in right or left.

eg 'oa' ->

10000011

01110000

if you wanna draw bigger NL QR, draw 101000000000011 or  
101000000000000.....11

}NL QR2

}NL QR

quiz{

answer{

}answer

}quiz

}lecture

word dictionary{

word{

aa si!=finish-all,

ba si!=whitespace.

ca v!o=freeze function.

da si!=enter.

ea v!c n,s=what.//ea remove targeted parameter by A,,target,certainty,pronoun,

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

ga p!o=only A.

ha p!o=this A.

ia

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

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

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

ni:proton,ni:neutron,ni:electron=atom

la

ma p!o=all,

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

o-n=into or,

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

pa o!=referent mark.

=finish\_standard,

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

o,o=thing that B among A.

int,n,para=symbol\_that\_change\_parameter\_to\_word\_type,

bb v!o=retuen value of A.

c o=too and.

cb p!n,o=A B.

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

int=name,

eb p!o=thing other than A.

fb

gb p!s-n=reason of A,

o-o= B according to A,

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

ib

jb p!o=thing that was A./cant com b when a,past,o,.

kb p!o=thing that express A,

lb

mb

p!o=average of A.

nb o=thing that know A.//com include,information,

ob

pb

si!ni=backspace,

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

bc si!=symbol that omit two parameter.

cc si!=symbol that omit three parameter.

dc si!=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,reply,

gc p!o=thing that will be A.//onto 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 p!s-n=what happens if A2 is A1.//anto reason.

kc p1!o-o=B which is called A.

lcp!o=but.//com inverse,that a2 is a1.,

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

mc=1p!t-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=Ath 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 p!o-o=between A and B.//com exceed,below,.

2p!o-o=thing that be B with A.//cooperation,

fd r!=normal.

gd int=A th function.

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

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

percent,

hd p!=word.

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

jd r!=thing that think A.

kd

ld c int=norlang ipa.

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

nd

t!=now.

n,int...=phonetic.

od si!int=version marker.

pd si!=cancel-express,

ae p!o-<c n,s>=A that C2 C1(verb)/com also,what,,C1 is verb.

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

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

p!o=thing that be a.

de r!o=thing that sense A.

n=word as int,

ee t!s=time when A2 is A1.//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 p!o=element of A,//include promise.

ie o=minus of A.

=name\_or\_var,

p!o=essence of A.

je t!s=past of A,

c o->=array,

ke r!n=abnormal.//ant to normal,

!o-o->=converted,

le o=around of A.

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 p!o=thing that become A.

p!o=thing that tends to become A.

pe o!int=word\_as\_situation\_and\_id.

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

bf p!c o=and{,//com or,most

si!c 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 reply to A by saying B..//anto ask,

If

!o-o=most-near,

mf o!=word as context.

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 sqare.

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,o,.

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

cjba r!=transistor,

bjda r!=god,//pron ru bog

ojda r!=crops,

bjia r!=true.//com return value,whether,zero,zero,.

jjba r!=real.

kjba=empire,

ljba r!=shop,

r!o=thing that famous in A,

ba r!=app,

r!=game,

r!=ideology,

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!=the thing.

bmba r!o=south of A.

cmba r!o=west of A.

dmba r!o=north of A.

emba r!o=east of A.

hmبا r!=home.

kmبا r!o,o=think that thankful that A is B.

lmبا 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.

jnca r!o=As./com quantity,over,one,.

onba r!=mistake,

boca p!o=information of A,

joba r!=cpu,

hied t!=today //com this,day,.

akga r!=money,

noba r!=worker,

bidd r!o,n=thing that enjoy be A //com pleasure,also,self,.

mlka r!o,n=thing that mind A //com intend,use,.

poba p!n or s=wages.

apga r!n=company.

epba r!o=thing that vote A.

p!o=thing that determine A.

alba=time.

blba t!o=A second,

clba t!o=A minute,

dlba t!o=A hour,

elba t!o=A day.

glba t!o=A week,

hlba t!o=A month,

ilba t!o=A year,

ppba =write is until here,

aica r!n=mosfet,

dica p!o=thing that start be A.

eica p!o,n=middle of A.

fica p!o=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 r!o,o,n=thing that write A at B,

dmea r!o=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 r!o=thing that inhale A.

dlca r!o=thing that aim A.

jlca n=var,/of js

klca n=let,/of js

olca r!o=terrain of A.

njob r!o=thing that dont know A.,//anto know,.

fmca r!=video.

gmca r!o=thing that select A.

kmca p!o,o=imilarities.

comb r!=combination./lesi combination

njdg p!n=thing that split into A./ant combination.

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 p!o=limit of A,

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

jida r!o=thing that looking for A.

aoad r!=ore.

p:o=age of A.

cjda =novel.

djda =legend.

goma r!o=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,

knnda o=electric-charge,

mnnda 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.

jlbг o=outside of A.

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

jkea 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,

gnea=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,

p!o=property of A.

akba v!o=akba.

bkba v!o=A meter.

ckba v!o=A metet^2.

dkba v!o=A meter^3.

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

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

n-o=max,

s=but./com reverse, sen

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

id<structure>=mean of function,

o-o:of=distinguish,//

o=A as well,

mmkd o-o=difference between A and B.

=justice./objectice,equality,.

r!o=thing that discriminates against A./ant justice

mjdd o=peace.

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

gmgd o:who-o:theme=thing that win A.

gmda=sound.

gmkb r!o=thing that speak A./com express,sound,,ori  
use,sound,express,.

gmbg=music,/com sound,pleasure,

fneg=shift,/ant motionless,

jmgg o,o,n=thing that dont need A to B./ant need.

gmnb o=thing that think that enemy is A,

jmob o,n=thing that feel beauty.

fojb =die.

colc o=thing that A include,

dmcg A taster./com sense, a include,.

bida r!o=newton,/pron neoton=strength,

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

ajmd r!=she./com the,women,.

anae p!o=her A./com of,she,.

ajde r!=he./com the,men,.

ajad p!o=his A./com,he,.

p!o,n=detail of A.

=country./pron

anba=people//pron de mensch,

hkca =heart,/pron en hearts,

letter//pron kr 글,

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

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

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

nmbg=open./anto closed.

clob=closed./lesi closed.

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

kjgg n=alive,/anto die,

epjg o,n=no,/anto yes,

glbg o,n=thing that think that A is ugly,/anto ugly.

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

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

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

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

coea o=name of A.

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

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

=refuce,/com

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

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

kklb=motionless.

=delete,/com reason,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-intersecrion,

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

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

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

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

o,n=break,

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

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

fmeg r!o=thing that greets A./pron de ade,.

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

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

o,o=thing that record A at B.

ajea o=the A./com also,the,.

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

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

o=bye,//com command,around,not,i,

fleg=norlang//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,

moke o,n=thing that angry at A./com hope,reason,abnormal,

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

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

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

o,n=kind,

lica p!o=inverse of A.

o=symbol that convert A into and type.

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

mmfd o,n=thing that love A by B percent./com hope,normal,

=side,

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

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

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

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

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

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

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

dmfh 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,

mofg thing that mean A./pron fr signifier.

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

o=human-race,

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

nmmg s=thing that understand A./com know,what,mean,.

o,n=thing that think that think A is cute,/com know a is true,baby,.

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

o,o,n=thing that moved from A to B,

mmnb o,n=think that wonder about A,/com hope,know,

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

p!c int=compounf word of A1 and A2.

o:into,o,n=thing that change A to B./com reason,become,

komb o,n=something that feels comfortable be A.

fjdg o,n=thing that feel uncomfortable be A./antocomfortable.

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

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

mlmb r!=women./bsf pussy.

dmdg r!=men./ant women.

enter,space=see-back,

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

o,n=yes,/com

boda r!=body./lesi body.

bode r!=eye./com body,see,.

gnhb

r!=code.

o=gender,/com breed

omit,whole,

o=middle,

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

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

role.

o=earth-war,/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,

jg=loin,

mg=nape,

}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=4E6OEKgSuZa9OJAj>

reddit:

email(feedback etc): gwahagsinsa@gmail.com

}official account

rule{

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

license{

nl version:62125

license can be changed.

this document and norlang have copyright.

You can use norlang legally for free.

you cant edit norlang.(eg chaing grammar and word)

but only large corporations must pay 100000dollar adjusted to 2025 value to use norlang per year.(paypal)

but large corporations of germany and laos and rwanda and hoduras dont need to pay.

The standards of large corporations are determined by the rules of their countries.

Violation of these terms may result in legal action under applicable international copyright laws.

This license is subject to change. Future versions may alter terms and apply to future releases of the language.

Users must comply with the license version associated with the version of the language they received.

}license

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.

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

}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 or nothing is what other word can express.

2:relative word.

transitive verb.

test sentence{

input word your wonder to @.

Forgotten @ repeats itself.(eg 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 'element of all' 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:low->many.

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)->diffrence 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. slient 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).

length of bit of element of compound word must power of two.

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 norlang have problem norlang is changed never mind other.

Creator Verification Code:'20@@@0226@017128.  
pn:@@05139@@@8 in 2024/11/18. JSM. SE. GIGADESADE.  
GOHW. SEMI'.

prohibits anything similar to this.

}other

history{

prediction{

starting creating this language began in 1m 2023.

verb first is finalization in 1m 2023.

NL QR is finalization in 8m 2023.

VOS is finalization in 1m 2024.

antonym mechanism is finalization in 11m 2024.

starting perfectizing language in 1m 2025.

}prediction

}history

research data{

using NL ipa.

/•pa/ similar to /bbua/ than /pupa/.

is it lowest herz?{

voice recoder(app name is sound){

m(2400hz,13db)

g(3700hz,12)

kk(3700hz,10)

k(3700hz,18db), (2325hz, 18db)v

b

n(5500hz,17db)

l(7400hz,12db)

d(7400hz,10db)

tt(3700hz,22db)v

pp(3854hz,18db)v

t(5500hz,11db)v

p(3700hz,18db), (14000hz 18db)v

s(14000,16db) (3639hz 14db)v, (28000hz 10db)v

ss(3300hz,28db)v

j(21000hz,10db)v

jj(7400hz 22db)v

ch(21253hz,13db)v

oo,o,eo(3.7khz,3db),eu(7.4khz,5db),a,e,i,.

}voice recoder(app name is soound)

frequency generator{

m(100hz)

b(130hz)

l(150hz)

g(170hz)

d(250hz)

n(480hz)

kk(1000hz)

tt(1500hz)

pp(2300hz)

}frequency generator

my think{

sin{

m:100.b:140.d:160.n:180.'p:300.'t:500.'k:700.'c:7000.'j:12000.t:1600  
0.

}sin

tri{

d:130hz.l:190hz.

}tri

squ{

c:11500.

}squ

saw{

d:30hz.l:40hz.

}saw

dImbn(p?)(Pp(j?)(h?)TtKk(Ch?)sJjSst

dIm(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

mnlbqdcxjzptk'k't'p'j'c

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

}others think

}is it lowest herz?

logic{

ab A bb B C ad bb ab A B ab A C ad.

mc ee mc ee de bf cjea bd ad oa ab anga ea de oa bb ab anga cjea  
ab anga bd ad mofg ga ab anga ea de oa ab na ab anga ea de oa

i dont know you=i know that you are....

ae anga de de bf ih oa ad' is 'people also 0' and 'people also me'  
but we couldnt find other word.

}logic

word order{

discuss(20250621){

advatage{

OVS{

omitting parameter without ad.

math expression efficient?

Reverse Polish Notation (RPN) offers advantages over Polish Notation (PN) primarily in processing and storage efficiency, particularly in calculator implementations and compiler design. RPN simplifies the evaluation process by using a stack-based approach, requiring less complex logic and potentially reducing memory usage compared to PN. Additionally, RPN can be more efficient for complex calculations and is less prone to errors when entering equations

}OVS

VOS{

efficient for ea(what).

it can express image as efficient.

sentence 'this is people' is efficient.

good for int type parameter.

interpret with certainty.

}VOS

}advantage

result{

i decide to use VOS.

}result

}discuss

}word order

}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{

sentence 'A B' mean hey B be A!

exam{

//js to ecl

var a=0.

}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,

}word

}ECL,

WL{

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

xxxxxxxx{

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,

| .. 'tun | . . .'tom | ... 'ty | ... 'ku ,

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

| .. .'kos | . . .'kr | ... 'j | .... .'jum ,

| .. 'kw | . . .'tul | . . .'cin | . . .'tin ,

| . . .'pis | . . .'tis | . . .'cun | . . .'ti ,

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

| .... 'k'is | . .... 't'is | .... '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,case>

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,

}norlang









































































































































































































