UNITED STATES POSTAL SERVICE

CUSTOMER'S RECEIPT

KEEP THIS RECEIPT FOR YOUR RECORDS

PAY TO:

ADDRESS

C. O. D. OR USED FOR

SEE BACK OF THIS RECEIPT FOR IMPORTANT CLAIM INFORMATION

NOT NEGOTIABLE

SERIAL NUMBER	YEAR, MONTH, DAY	POST OFFICE	AMOUNT	CLERK
04407299752	2002-09-16	809030	$25.00	009

UNITED STATES POSTAL SERVICE

CUSTOMER'S RECEIPT

KEEP THIS RECEIPT FOR YOUR RECORDS

PAY TO:

ADDRESS

C. O. D. OR USED FOR

SEE BACK OF THIS RECEIPT FOR IMPORTANT CLAIM INFORMATION

NOT NEGOTIABLE

SERIAL NUMBER	YEAR, MONTH, DAY	POST OFFICE	AMOUNT	CLERK
04407299763	2002-09-16	809030	$10.00	009

Cover

To: Donald W. Kerr
Deputy Director of Science
and Technology.

Re: Plasma physics.

14

(Faxed to Dr. Q-Tel (703) 248-3000
9/28/02
24hr 3001 fax)

Thank you for your submission.

Over the coming weeks we will be evaluating the proposal submission. If we need more information/documentation, we will be in touch.

Please be assured that we respond to each and every proposal. You will hear back from us.

In-Q-Tel, Inc.

David Merrill
[(719 [redacted]

Disclosure Document Program
Box DD
Commissioner for Patents
Washington, DC 20231

Registered Mail # RA 028 358 022 US

Dear Commissioner;

I am the inventor of the disclosed invention. I have enclosed a money order for $10. File the document under your Disclosure Document Program protocol.

Additionally, I have enclosed a money order for $25. Send a copy of the invention as filed with true and correct certification and proof of filing in your office on its face to:

La Vell Investments
2200 Maywood Avenue
San Jose, California.
[95128]

La Vell = Secret proprietor
of Homeland Security Research
Corporation = Victorian Hair
Salon = CIA front organization
to launder appropriated funds
from suspected "terrorist"
accounts into HSRC. Problem:
no Congressional or judicial
oversight!

Thank you,

[redacted]
David Merrill

David Merrill
[redacted]

Following is the basic infrastructure of an invention I developed in 2000-2001 while developing utilities in advanced-resonance inductive plasma physics. Aside from weeding out vulnerabilities in compressed information infrastructures as a viral tapeworm, this same program translated into BASIC will easily test supercomputers before putting them to parallel and quantum running of genome sequences:

> ...whose solution determines the average result, also termed expectation value, of every conceivable experiment on the physical system under examination... **Schrödinger's wave equation** from the *McGraw-Hill Concise Encyclopedia of Science and Technology* 2d.

By varying the common paradigm on the Spearman-Brown Prophecy Formula we find applications in prior knowledge psychometrics - forecasting.

The following disclosure document is contextually designed to express the application which on March 14, 2001 caused the United States Corporation to post a general bond for twice the national debt. I cannot extract this from the mathematics without losing the expression of applicability and then again encrypting the technology further will probably render it even more unintelligible to the reader without me teaching the technology in a seminar forum.

In the United States District Court
for the District of Colorado

U.S. DISTRICT COURT
DISTRICT OF COLORADO
2001 MAR [illegible] A.M.
JAMES R. MANSPEAKER
CLERK
BY [illegible] DEP. CLK

Spoken for by of the family Petitioner in response to NOTICE AND ORDER and STOP WORK ORDER of Respondent	**01-D-C** **Docket summary and state court prayer** -In Admiralty- Re: alleged compulsory forfeiture of rights, privileges and immunities in the original estate, a.k.a. 'common law'.

According to law Metro organization was duly served notice and summons on 2/8/01 to appear and answer the charge of placing a notice of lien into dominions and possessions of without *mesne* due course in admiralty - properly filing first in the appropriate district court. The United States according to the First Judiciary Act, September 24, 1789 has jurisdiction of these matters and according to the 'saving to suitors' clause of same Act, within the scope of providing remedy for There are no events historical or otherwise which alter or abrogate 's absolute and authoritative right of visitation. After service of summons, fifteen (15) days by the calendar, on 2/23/01 checked with deputy clerk in district court to find there had been no response or issuance of process by Wiley Young - the federal judge appointed by the district court clerk to this matter of remedy. Default and declaratory judgment was formally executed at that time. On 2/ /01 the City agent City of Code Enforcement attempted to state answer by taping papers to s front door and on 2/ /01 an attempt to answer was made for agent Regional Building Department by private process server at 's home. Both of these attempted answers are found in the district court case jacket. On 2/ /01 the County Tax Assessor joindered itself with Metro organization's seizure cause by posting a doorknob notice on 's front door.

Immediately, 2/26/01, the County assessor was joindered by authority. On 2/ /01 a private process server handed papers that expressed Metro organization's regret that process had been overlooked at time of judgment day. On 3/ /01 a provisional extension was granted stipulating answer to the charge might be filed by 3/14/01 but the judgment would stand unless Metro could prove it had actually properly filed the *mesne* process in district court prior to issuing any notice of lien. On 3/8/01 two agents were making forcible reentry on Metro's alleged claim by poking around 's possession by threat of "ordinance" (artillary). A petition to the district court was immediately filed, served by mail 3/13/01 requesting injunctive relief in admiralty. On 3/16/01 Metro agents handed inert process directed toward a magistrate. All benefit of discussion is refused for cause and judgment executed sits *res judicata*. On or around filing this docket summary checked the case jacket in district court and found no satisfactory answer to justify notice of lien or trespass against . Metro organization has failed to answer this counterclaim and the district court is greatly appreciated for the remedy provided in the due course of admiralty according to international law and the Holy Bible providing competent judiciary and common law by the will of God. This summary and explanation of the foundational state court prayer were entered into the case jacket at the timestamp above.

Prayer

Standing on presumption of a numero-linguistic standing wave extant between God the creator of the Universe and man in the information thereof one can develop a resonant prayer. Ancient herbal remedies and modern pharmaceuticals take advantage of vibration resonance and cancellation by cyclic representational waveforms which may be easily reduced to mathematics (example: Tesla coil based Bioenergy Spectrum Analysis). This is more than theoretical physics. *Numerology* by E.T. Bell, Williams and Wilkins Co. 1933, explains the injury Pythagoras inflicted upon humanity by developing numerology instead of arithmetic at mankind's formative intellectual cusp. Albert Pike explains the importance of numerology in Freemasonry on and around p.841 of *Morals and Dogma*. Bell reasonably speculates without numerology the Industrial Revolution might have occurred well before the time of Jesus Christ. An interesting note extracted from page 65 (Bell quoting *Life of Pythagoras*, Iamblichus, Chapter 4) states Pythagoras was captured from Egypt into Babylon by Cambyses for twelve years of his life giving Pythagoras the opportunity to study under the Magi of Chaldea. This would have occurred sometime around 550 BC. Also, if the historical account is correct in both the Bible and Iamblichus, Pythagoras studied under the chieftain of the Magi and famous prophet Daniel. See Daniel 2:48. It is likely the Magi

of Matthew 2:1-12 were commanded to posthumously bring gifts and homage to the Messiah by Daniel. Daniel, with his gifted command of astronomy informed the order to watch for the alignment of Regulus and Jupiter[1] on Tishri 1, September 11, 3 BC allowing them two weeks - Leviticus 23 - to get to Jerusalem for the Feast of Tabernacles.

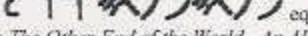
Events in the life of Daniel were amazingly accurate numerically. MENE, MENE, TEKEL, UPHARSIN[2] converts to 2,520 gerahs, the subject matter being gold weight. In 1887, The *Approaching End of the Age*, Dr. Grattan Guinness[3] predicted Jerusalem to be conquered by a Christian power in 1917 on this information and Nebuchadnezzer starting conquest of Israel in 604 BC. General Allenby took Jerusalem as predicted by Guinness, without a shot being fired. Further examination, Daniel 5:31, shows the event of the Chaldean overthrow, the "Writing on the Wall", was in 524 BC. Counting 2,520 years brings us to 1996-97; the moment of the 31 Day Government Shutdown which occurred simultaneously with United States District Court case 95-D- . The conveyance of assets was sluggish and complicated by a Chinese promissory note delaying default until about early March 1997; (briefly) considered a soft money campaign contribution (Coffee Tapes) to Bill Clinton. Turn-around Tuesday happened directly after western ports shut down to Chinese vessels, disrupting the foreclosure process and reneging the Chinese note; crashing the Asian Stock Market. See Associated Press releases on 1/23/96 *Rubin predicts default date* and 10/18/97 *Reno asks Chinese aid in finance probe*.

The inquisitive rabbis have understood an intrinsic relation between physical reality and the 22 Hebrew consonants and numbers - the perpetuation of Creation through God speaking the *debar* - the Word. See John 1:1 as correlated with Genesis 1:1. Also, read John 7:15 (Authorized King James Version) in context. Jesus knew all about the rabbinical practices and without esoteric initiation. Freemasonry and other esoteric societies capitalize on "32", the 22 consonants and 10 digits (3+2=5; the Pentagram and Golden Mean).

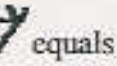
The Hassidic Jews point out that Eliezar spells "318" and in Genesis 14:14 Abram's commander-in-chief is referred to by number as his name - *nom de guerre* - like the all upper

---

[1] Skyglobe software developed by NASA - Skyglobe 3.5 shareware, Mark A. Haney, 1993. ClassM Software, 284 182nd Avenue, Caledonia, Michigan [49316].

[2]  equals 1126 equals 10 equals 1. One is generally associated with Unity

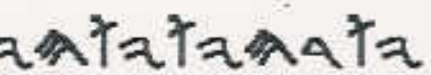
[3] Excellent reading is *The Other End of the World - An Alternate Theory Linking Prophesy and History* by Roger Rusk, Plantation House 1988. Guinness is discussed on page 116 and page 115 says, "God knows all about arithmetic. He is no stranger to algebra, geometry..." Also an appendix is called *Calendars and Time Cycles*.

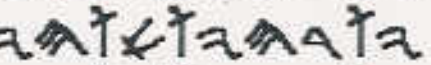
case description of name[4] found today on the commercial Driver License; commercial *engagement*. CAFR, Comprehensive Annual Finance Report, relevant to this cause is "ransom" in Exodus 30:12. This tax is collected as compulsory atonement for wartime atrocities committed by the *nom de guerre* warriors. *Kophar*  equals 300 equals 3 which means *Completion*. God is spelled in a unique fashion in three verses of Isaiah; the Protector - 41:13; the Saviour - 43:3; and the Redeemer 48:17. Spelling   equals 153 and by rabbinical claim corresponds to the 153 fish in the net at John 21:11. Summing the numeric counterparts of Biblical Hebrew words found in *The Word; The Dictionary that Reveals the Hebrew Source of English*, Isaac Elhanan Mozeson, SPI Books New York, 2000, or *Strong's Exhaustive Concordance* or *Young's Analytical Concordance* shows the remarkable consistency between word meaning and number meaning. These mental associations are however subjective and a convincing list would be too long for this explanation.

Only these few classical examples of numero-linguistics are mentioned in an effort to briefly explain the prayer into state court. The reader is free to dismiss as much or little at a whim. The state court into which this case is remanded is Jesus Christ of Nazareth who is the living God. That is to say by name:

Yehoshuah H'Natzrith H'Mashiach asher Yodi-hehe-wavoh-heyah.

Yehoshuah=1 

Yod-heh-wav-heh= 72 

Yod-heh-wav-heh= 63 

Yod-heh-wav-heh= 45 

Yod-heh-wav-heh= 52 

(Tetragrammatron) 232=7

Ruach Kadosh=12=3 

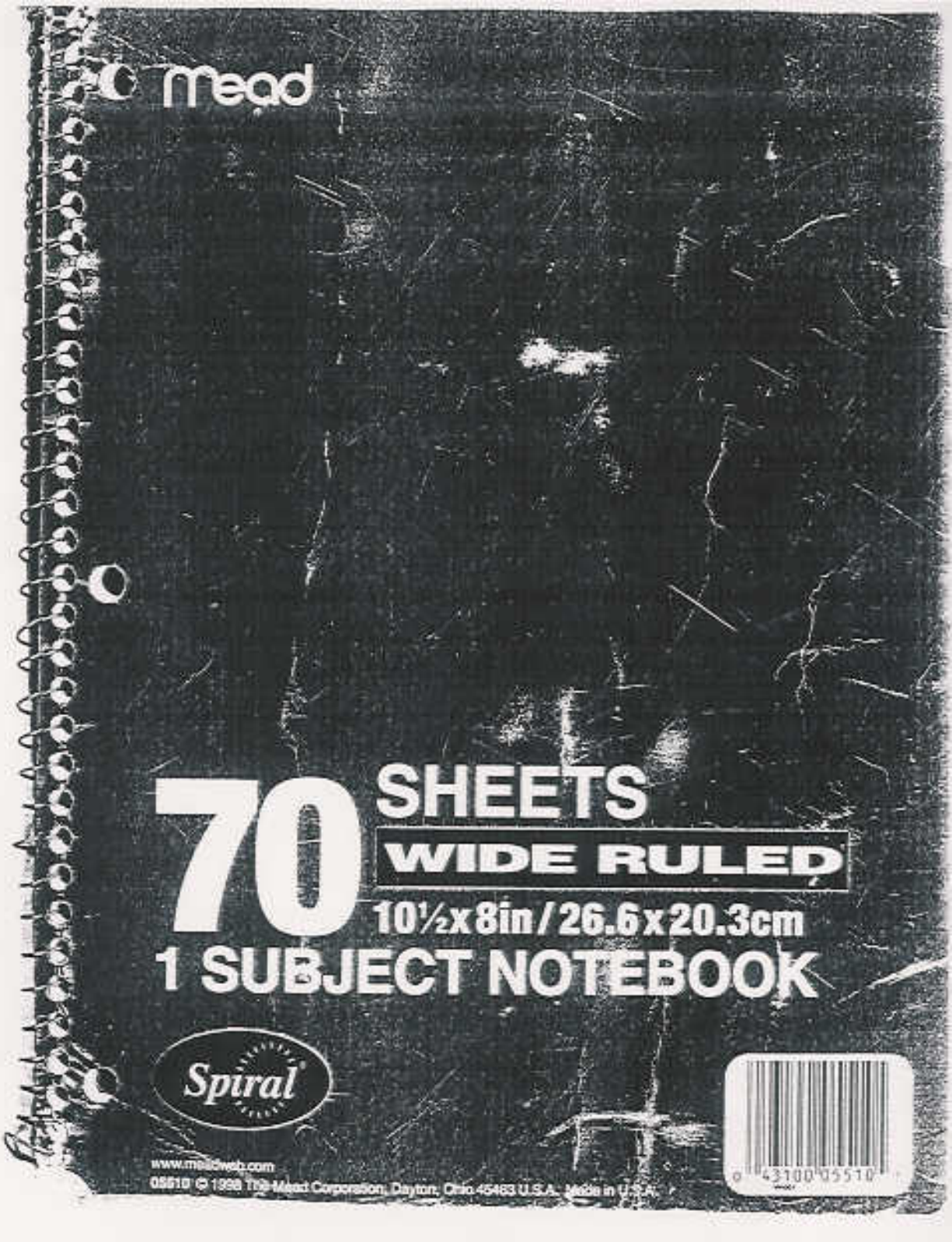
(Holy Spirit)

[4] This invention is called *quasi-mathematica* and attributed to the Roman Catholic Church. See *The Metaphysics of Saint Thomas Aquinas*, and *The Just War in Aquinas and Grotius*, by Dr. Joanne Tooke.

and this is the simple intent of this state prayer, to call on God by name with the cause expressed in the petition and counterclaim. A matter of consanguinity and inheritance in the original estate.

An effort has been made to develop resonance upon which to base truly effective prayer. The spiral notebook used for program development (*HP-42S* programmable calculator) is attached for a literally mathematical explanation.

The "Great Fall' in Christianity is when Adam lost communion with God for disobedience. Anthropomorphic symbolism explains the left hand of God, Mercy or Grace was skewed by 'evil intent' - Adam's lack of repentance - and His right hand, Power, compensated by showing a severity called Judgment. The Book of Judges is a classic example of how God uses Israel's enemies to judge and correct Israel. The computer programming is executed for the purpose of resonating the standing wave and preparing the way for the return of our Lord and Savior Jesus Christ like the prayer of Isaiah 40 and in the spirit of the prophet Elijah found in John the Baptist at the first advent of the Messiah. In Hebrew, this is *tikkun*. A close translation in English is *restoration*. Amen.


Mead
70 SHEETS
WIDE RULED
10½x8in/26.6x20.3cm
1 SUBJECT NOTEBOOK
Spiral
www.meadweb.com
05510 © 1998 The Mead Corporation, Dayton, Ohio 45463 U.S.A. Made in U.S.A.
0 43100 05510

Example program p-173

reg 1 reg 2 reg 3

1 + 2 = 3

3 + 2 = 5

5 + 3 = 8

8 + 5 = 13

13 + 8 = 21

21 + 13 = 34

$\frac{\text{reg 2}}{\text{reg 3}} \times 100 = \text{reg 4}$

reg 5 — reg to Counter

STO reg 4

+ New reg 4

+ New reg 4

+ New reg 4

↓ # of iterations

X Register

Sum reg 5 ÷ x register.

Begin sequence at [1] (STO 00)

Number of Iterations? [100] (STO 01)

Fibbonacci step?

RCL 00

+ [1] (STO 02)

= 2 (STO 03)

RCL 00

RCL 03

+ [ ] (STO 04)

RCL 02

RCL 04

÷

100

X

[ ] (STO 05)

RCL 03

RCL 04

+

[ ] (STO 04)

x 19

y 2 — STO 00

+

0 3 — STO 01 (+1)

RCL 00 —

+ (-1)

5 — STO 00

RCL 01

+ (+1)

8 — STO 01

RCL 00

+ (-1)

13 — STO 00

RCL 01

+ (+1)

21 — STO 01

RCL 00

+ (-1)

34 — STO 00

RCL 01

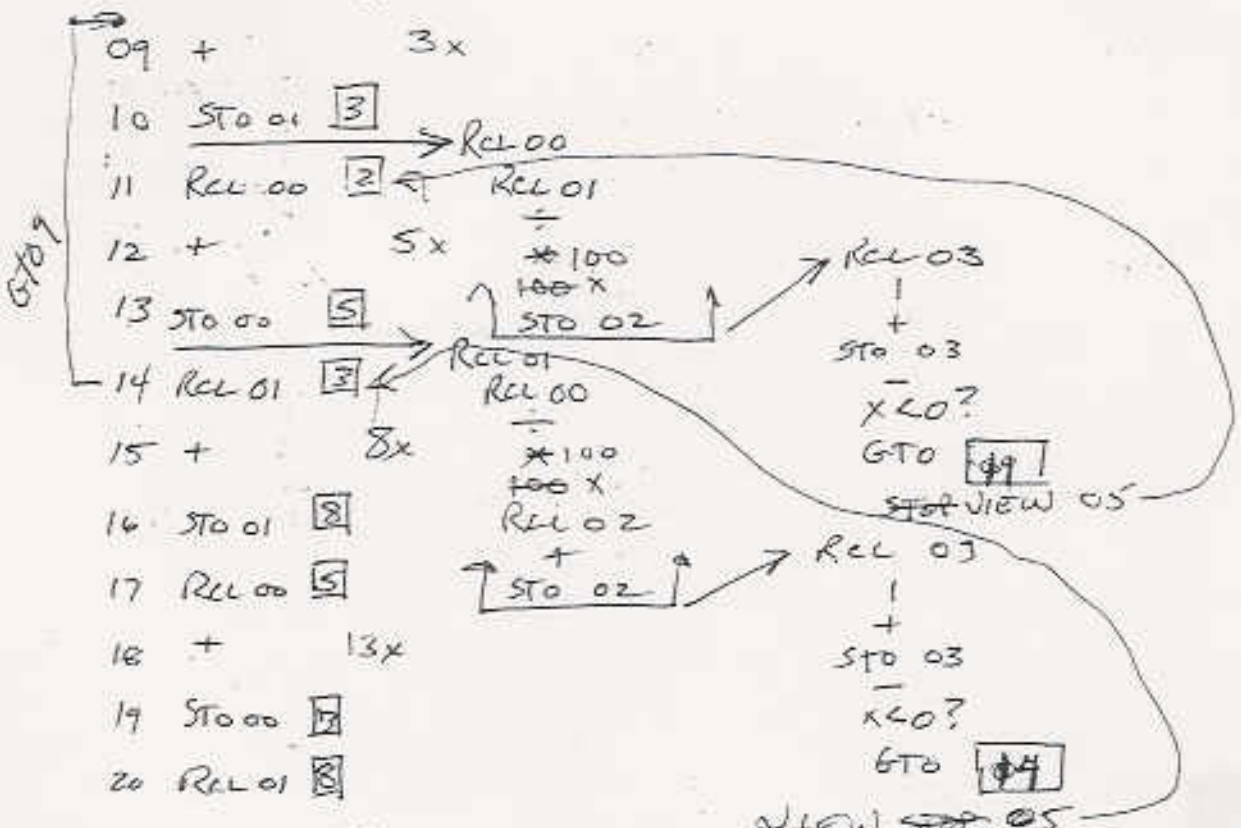
+

55

10

45

```
00
01  LBL "FIBB"
02  'ITERATIONS?
03  PROMPT
04  STO 04
05  1          STO 03
06  ENTER      X
07  2          ↑ 1X, 2x
08  STO 00 [2]
09  +          3x
10  STO 01 [3]
11  RCL 00 [2]
12  +          5x
13  STO 00 [5]
14  RCL 01 [3]
15  +          8x
16  STO 01 [8]
17  RCL 00 [5]
18  +          13x
19  STO 00 [13]
20  RCL 01 [8]
21  +          21x
22  STO 01 [21]
23  RCL 00 [13]
24  +          34x
25  END
```



~~CLST~~

~~CLRG~~

CLST LBL "FIBB"
CLRG ITERATIONS?

PROMPT 115

50 STO 04
STO 05 ~~CLST~~ ~~CLRG~~

STO 03

ENTER

2

STO 00

LBL 00		$\frac{3}{5}$	$\frac{8}{13}$	$\frac{21}{34}$
+	$\frac{15}{3}$	$\frac{-12}{8}$	$\frac{-10}{21}$	$\frac{-8}{55}$
STO 01				
RCL 00	$\frac{3}{2}$	$\frac{8}{5}$	$\frac{21}{13}$	$\frac{55}{34}$
RCL 01	$\frac{3}{.6667}$	$\frac{5}{8}$	$\frac{13}{21}$	$\frac{34}{55}$
÷		.6250	.6190	.6182
100				
X	$\frac{3}{66.6667}$	$\frac{8}{62.5}$	$\frac{21}{61.9}$	$\frac{55}{61.8182}$
STO 02				
STO+ 05				
RCL 03	1	3	5	7
1	$\frac{1}{1}$	$\frac{3}{1}$	$\frac{5}{1}$	$\frac{7}{1}$
+	$\frac{66.6667}{2}$	$\frac{62.5}{4}$	$\frac{61.9}{6}$	$\frac{61.8182}{8}$
STO 03		4	6	8

RCL 04	2/15	4/15	6/15	8/15
−	−13	62.5/11	61.9/−9	61.8987/−7
x≠0?	YES	YES	YES	YES

GTO 01
FIX 10
VIEW 05
STOP

RCL 05
RCL 04
1/x
+
÷

RCL 01
RCL 00
x<y?
x≥y
÷
100
x
INT
+

LBL 01

RCL 00	−13/2	−11/5	−7/13	−7/34
RCL 01	2/3	5/8	13/21	34/55
+	−13/5	−11/13	−9/34	−7/89

STO 00

RCL 01	5/3	13/8	34/21	89/55
RCL 00	3/5	8/13	21/34	55/89
÷	5/.6	13/.6154	.6176	.6180

100

x	5/60	13/61.5385	34/61.76	89/61.7978
RCL 02	60/66.667	61.5385/62.5	61.7647/61.9048	61.7978/61.8182
+	5/126.667	13/124.0385	34/123.6695	89/123.6159

STO 02

STO + 05

RCL 03	126.6/2	124.0385/4	123.6695/6	123.6159/9
1/x				
+	126.6/3	124.0385/5	123.6695/7	123.6159/9

STO 03

RCL 04 3/15 5/15 7/15 9/15

– –12 –10 –8 –6 –4 –2

X ~~≠~~0? Yes Yes Yes Yes

GTO 02

FIX10 → VIEW 05 } RCL 05

RCL 01 → RCL 04 ÷

RCL 00 STOP }

X<Y?

X≷Y LBL 02

÷ RCL 01 3 –10/8 –8/21 –6/55 –4/144

100 RCL 00 3/5 8/13 21/34 55/89 144/233 377/610 2584/1597

X

X≷Y ~~RCL 02 166.667~~

GTO 00

..END.

4

5 60.4349816352
6 ~~166.66667~~ 60.6249422847
7 60.7740972264
8 60.8978367350
9 60.9796086176
10 61.0544237530
11 61.1168645676

10 61.0544
15 61.2542
61.4111
20 ~~61.4816~~
25 61.4739
100 61.7218

16 61.3187812
17 61.3187812
18 61.3697936768
19 61.3697936768

17,711
28,657
61.8%

100 iterations

RCL 00 61.7218295837 ← Cumulative ration average

RCL 01 61.8033988749 ← Current iteration ration

1973 - Stanley Cohen invents gene splicing - 3/12/01 AP Israel outlaws human cloning - Ashkenazim are for cloning; Sephardim against it. GT p. A6

The pentagram is a Laplace transform, Fourier series of the extensive memory - double helix.

⇓

☆ ⇒ Golden mean.

OPEN

↓

STORE ITERATIONS

↓

FIRST ITERATION AGE - 50

↓

LBLOOD

ADD FIBB

↓

DIVIDE # X 100

↓ — ADD TO % TOTAL

INCREMENT COUNTER

SUBTRACT ~~COUNTER~~ ITERATIONS FROM COUNT COMPARE TO ZERO

↓

NO

DIVIDE ~~DISPLAY~~ SUM OF PERCENTS BY ITERATIONS & DISPLAY

Yes → RECALL FORMULAS → ADD FIBB

RECALL & ADD FIBB

↓

DIVIDE # X100

↓

ADD TO % TOTAL

↓

INCREMENT COUNTER

↓

Subtract ITERATIONS FROM COUNT COMP TO 0

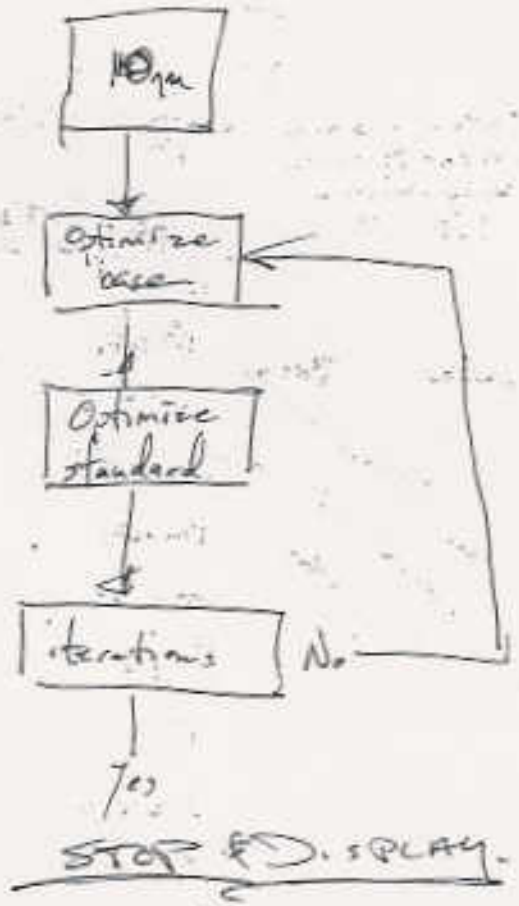
↓

NO

DIVIDE SUM OF PERCENTS BY ITERATIONS & DISPLAY

Yes → RECALL FORMULAS → ADD FIBB

	atomic	atomic weight
Chlorine Cl	Z = 17	35.453
Carbon C	Z = 6	12.01115



isotope number base

61%

Pentagram ☆ or Golden Mean is a Laplace transform and Fourier series of the Double Helix.

Neutron
1.00866520

− .00866520
1.00000000

STO REG

^1H	01	~~1.00866520~~ 1.0078251[illegible]
^2H	02	~~1.007~~ 2.01410222
^3H	03	3.01604971
	04	
	05	

+1 → 01
Δ 02 (X>Y?) if current isotope = current isotope skip routine 1 iteration
÷ dec places?
STO 103+
01
02 (+1) 03 — count passes
÷ dec places?
STO 103+
⋮
RCL 103
RCL 104 count passes
÷
STO 105 (average decimal places)

Atomic weight entry
~~4BL~~ LBL "STD"
0 STO 00
"H=?"
PROMPT
BINM
STO 01
"He=?"
PROMPT
STO 02
"Li=?"
PROMPT
STO 03
"Be=?"
PROMPT
STO 04
⋮

use storage ~~formulas~~ not registers

1408 lines
−147
1260

MVAR
Manipulate 100 elements.

17 BYTE

RCL 01
RCL 02 PRECISION 02
ALL full precision
RCL 01 BINM
RCL 02 BINM
Y > X ?
÷
X ≷ Y
÷
(CLRST)
(CLR R)
RCL 01
RCL "INCR"
X ≷ Y ?
÷
(. = 46)
(1 = 48)
X ≷ Y
÷
LBL 00
ATOX
X ≠ ?
GTO LBL 00
RCL ST X
46

LBL "TST"
2.5011
ARCL ST X
VIEW
END.
RCL 01
RCL 02
÷
X≷Y
÷

ATOX

ATOX
X = 48 ?
48
↑ enter
ATOX ?
X = Y ?
GTO 00
STO 99

47155304284

Find decimal

Add up Zeros after Decimal place

REGS Atomic weights
01 - 99

ELEM Current weight
01 - 99

INCR Test Weight (excluding ELE)
01 - 99

ACCRL Wholeness (0's after dec.)

```
0
STO 00
DATA
CLRST
CLRX
LBL 03 →
RCL 01   RCL "ELEM"
RCL 02   RCL "INCR"
x≠y?  CF 00
x<y?
GTO 02
÷
x≤y?
÷
ARCL ST X
LBL 00
ATOX
X≠y?  46
GTO 00
LBL 01
ATOX
#8
X≠y?
GTO 01  SF 00
LBL 02
STO+ 00
STO "INCR"
GTO 01
TEST  FC? 00
GTO 03
```

1
STO "ELEM"
STO "INCR"

Golden mean/liner grading

Fibonacci
61.554% @ 32 iterations

38.5% Grandpa
Dad / Grandma
50%
50% Grandpa
Mom / Grandma

Contributions increase → exponentially
but effects diminished by time ←.

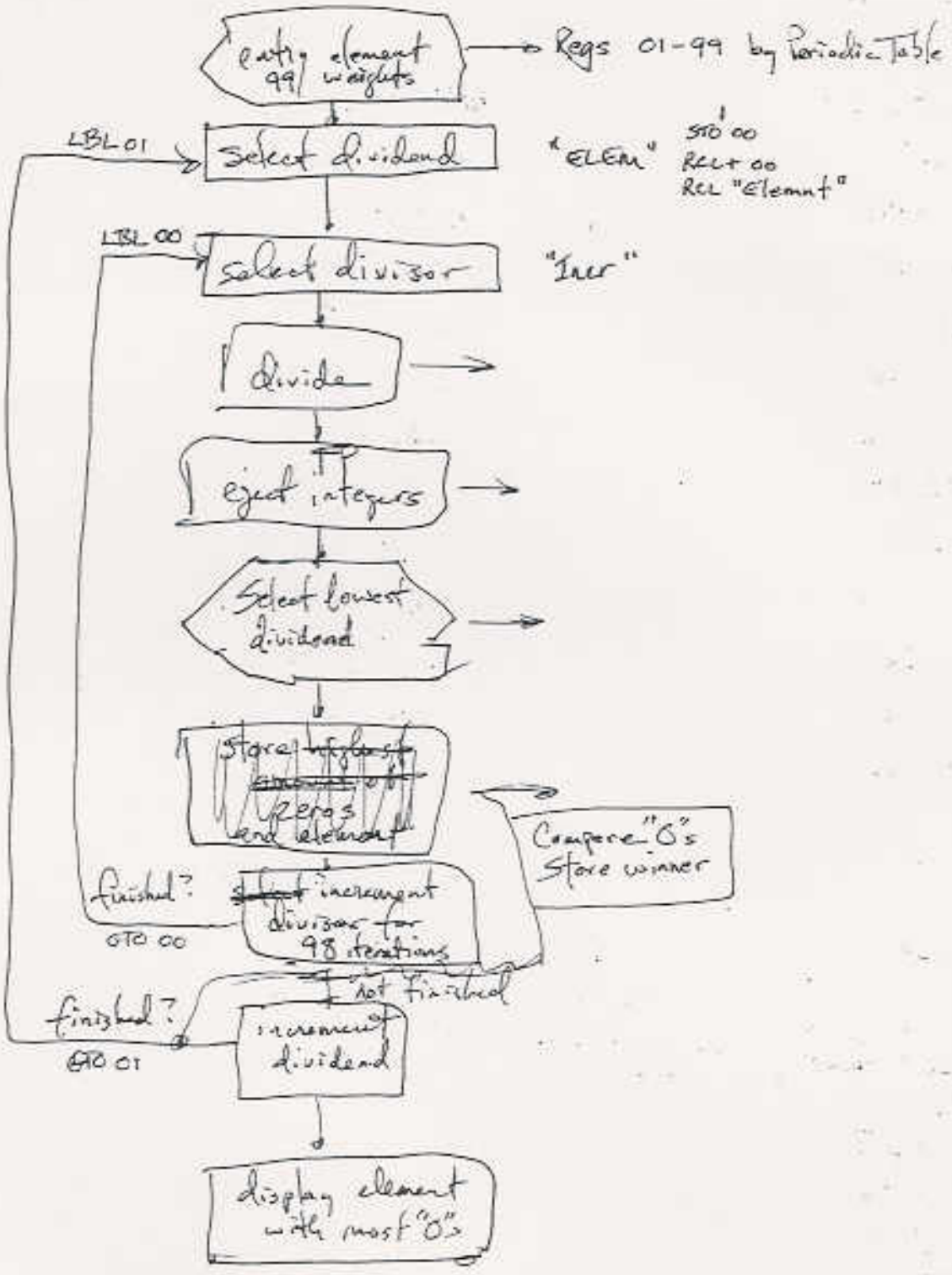
Standing wave
quasi-linguistic interface

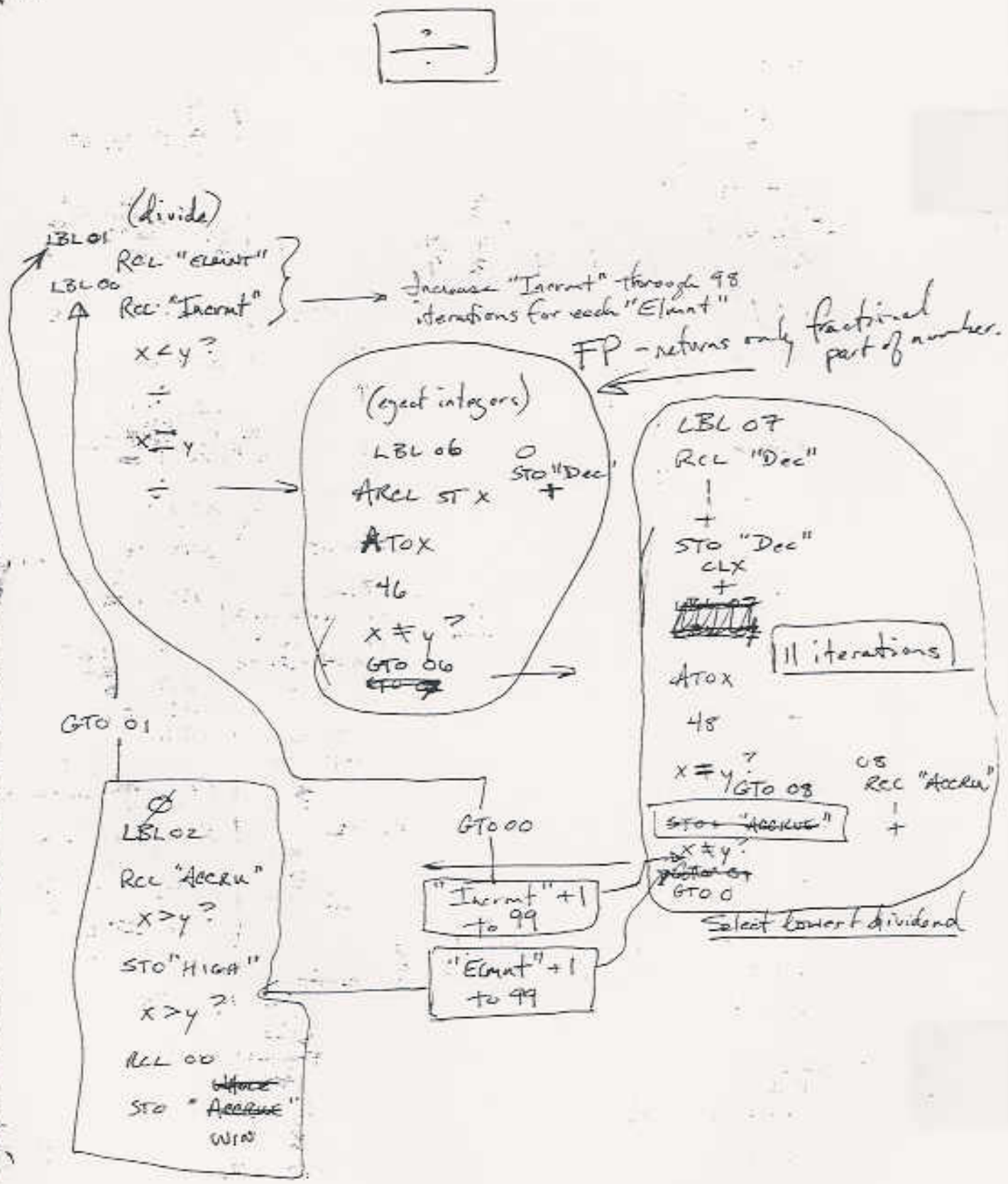
LBL "STRD"
1
STO 00
~~"T"~~
~~STO "ELEM"~~
~~STO "INCR"~~

~~LBL "STOD"~~
"H?"
PROMPT
STO 01
"He?"
PROMPT
STO 02
"Li?"
PROMPT
STO 03
"Be"
PROMPT
STO 04
.
.
.
CLRST
~~CLRA~~ LBL 03
~~RCL 0~~ RCL "ELEM" ~~RCL ST X~~
~~RCL 01~~ RCL "INCR" ~~RCL ST X~~
CF 00
x=y?
GTO 02
x<y
÷
x≥y
÷

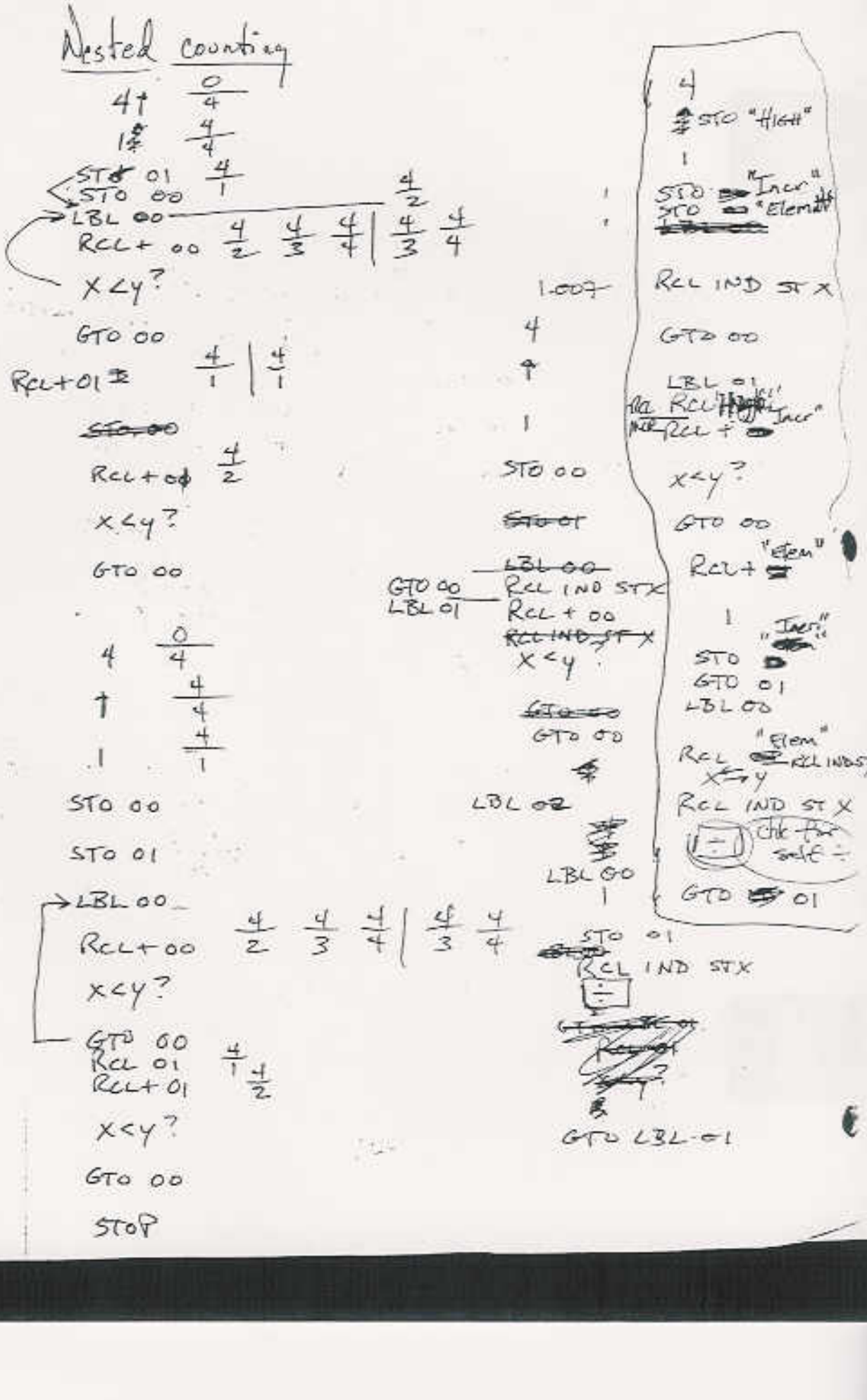
≠
RCL 00
RCL IND ST X

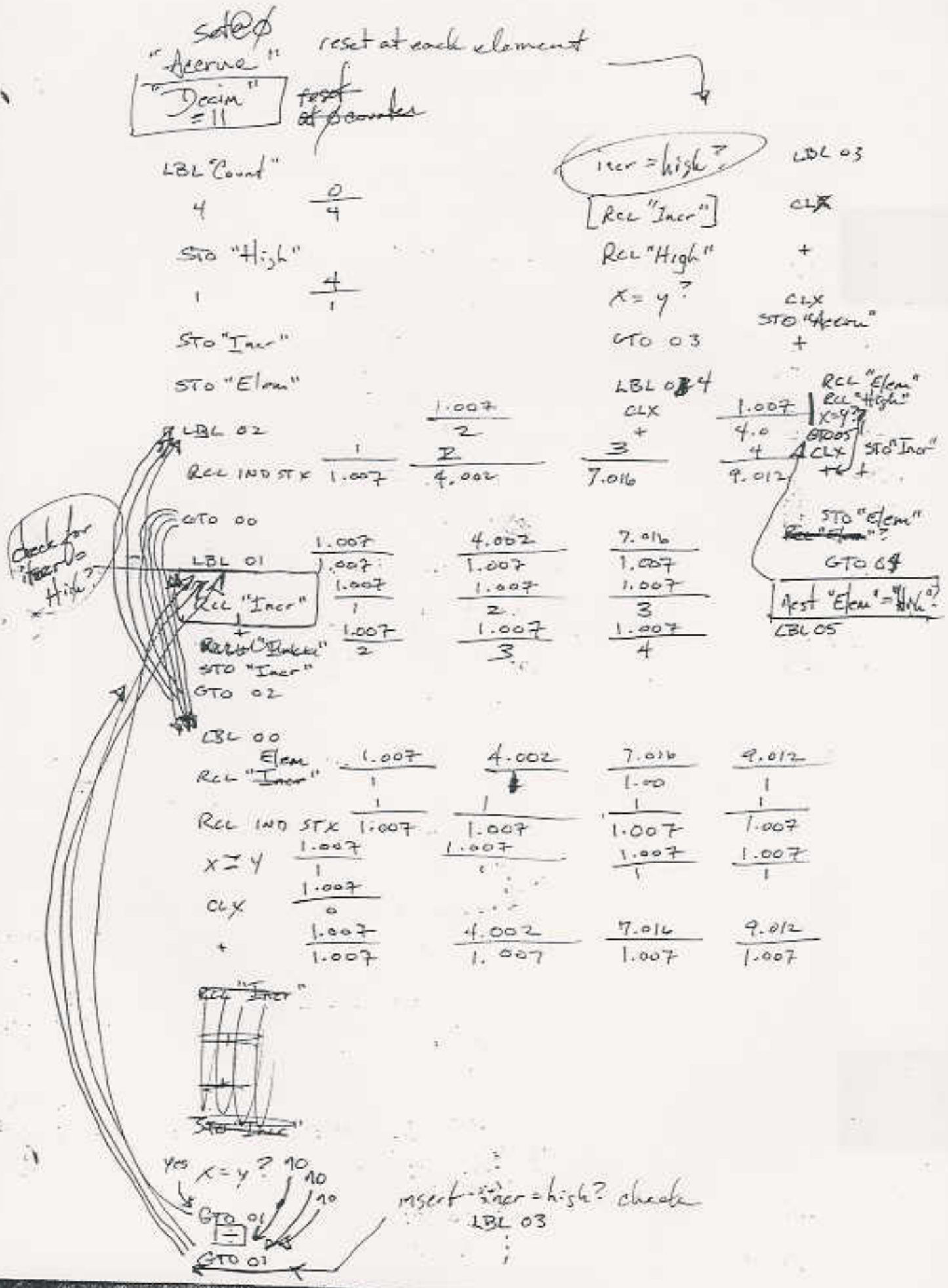
ARCL ST X
LBL 00
ATOX
46
x≠y?
GTO 00
LBL 01
ATOX
48
x≠y
GTO 01
SF 00
LBL 02
~~STO+ 00~~
STO+ "INCR"
FS? 00
GTO 03
GTO 01


entry element 99 weights
Regs 01-99 by Periodic Table
LBL 01
Select dividend
"ELEM"
STO 00
RCL 00
RCL "Elemnt"
LBL 00
Select divisor
"Iter"
Divide
eject integers
Select lowest dividend
Store zeros and element
Compare "0"s Store winner
finished?
GTO 00
increment divisor for 98 iterations
not finished
finished?
GTO 01
increment dividend
display element with most "0"s


(divide)
LBL 01
RCL "ELMNT"
LBL 00
RCL "INCRMT"
X<Y?
÷
X⇄Y
÷
Increase "Incrmt" through 98 iterations for each "Elmnt"
FP - returns only fractional part of number.
(exact integers)
LBL 06
0
STO "DEC"
+
ARCL ST X
ATOX
46
X≠Y?
GTO 06
LBL 07
RCL "DEC"
!
+
STO "DEC"
CLX
+
11 iterations
ATOX
48
X≠Y?
GTO 08
05
RCL "ACCRU"
!
+
X≠Y?
GTO 07
GTO 0
Select lowest dividend
GTO 01
Ø
LBL 02
RCL "ACCRU"
X>Y?
STO "HIGH"
X>Y?
RCL 00
STO "ACCRUE"
WIN
GTO 00
"INCRMT" +1 to 99
"ELMNT" +1 to 99

Nested counting




setup
"Accum"
"Decim" = 11
reset at each element
reset of counter
LBL "Count"
4
STO "High"
1
STO "Incr"
STO "Elem"
LBL 02
RCL IND STX
GTO 00
LBL 01
RCL "Incr"
+
STO "Incr"
GTO 02
check for "Incr" = High?
LBL 00
Elem
RCL "Incr"
RCL IND STX
X=Y
CLX
+
1.007
4.002
7.016
9.012
RCL "Elem"
RCL "Accum"
STO "Elem"
GTO 04
LBL 04
CLX
+
3
7.016
Incr = high?
[RCL "Incr"]
RCL "High"
X=Y?
GTO 03
LBL 03
CLX
+
CLX
STO "Accum"
+
1.007
4.0
4
9.012
GTO 05
CLX
+1
STO "Incr"
+
Test "Elem" = "High"?
LBL 05
RCL "Incr"
STO "Incr"
Yes
X=Y?
10
10
10
GTO 01
GTO 01
insert "Incr = high? check"
LBL 03

01 1.007825190000 H
02 4.002603120000 He
03 7.016003900000 Li
04 9.012185500000 Be

3.9715 He/H
6.9615 Li/H
8.9422 Be/H
1.7528 Li/He
2.2515 Be/He
7.0160 Be/Li

LBL "STND"
4 → 0/4
STO "High" → 4/1
1
STO "Incr"
STO "Elem"

LBL 02
RCL IND ST X → 1/1.007 → 1.007/2 → 2/4.002 → 3/9.016 → 4/9.012
GTO 00

LBL 01
RCL "Incr" → 1.007/1 → 1.007/2 → 2/1
1
+
STO "Incr"
GTO 02

LBL 00
RCL "Elem" → 1.007/1 → 4.002/1 → 7.016/1 → 9.012/1
RCL IND ST X → 1/1.007 → 1.007/1 → 1.007/1 → 1.007/1
X ⇄ Y → 1.007/1 → 1.007/1 → 1.007/1 → 1.007/1
CLX
+
X = Y ? yes → 1.007/1.007 → 4.002/1.007 → 7.016/1.007 → 9.012/1.007
no → GTO 01
no → ÷
no → GTO 01

RCL "High"
RCL "Elem"
X≷Y?
GTO 03

LBL 03
Recall & display
→ LBL 03

82 strokes

8.94221099991

is $\frac{9.012}{1.009}$ Be #

x<y?

÷

x=y?

÷ ——— EJECT & FIND SMALLEST

Rcl "High"

Rcl "Incr"

x≠y?

Gto 01

CLX

+

CLX

+

Rcl "Elem"

1

Sto "Incr"

+

STO "Elem"

CLX

+

Rcl "High"

Rcl "Elem

x≤y?

GTO 01

Display & STOP

```
LBL "std"
  99
STO "High"
   0
STO "Accu"
STO "Frac"
STO "Low"
 100
STO "Win"
 CLST
   1
STO "Incr
STO "Elem"
  LBL 02
 RCL IND STK
  GTO 00
   LBL 01
    RCL "High"
    RCL "Elem"
     X≥Y?
     GTO 03
      RCL "Incr"
        1
        +
      STO "Incr"
      GTO 02
      LBL 00
       RCL "Elem"
       RCL IND STK
        X≠Y
        CLX
        +
        X=Y
       GTO 01
       X>Y?
       X≠Y
        ÷
        FP
       STO "Frac"
       RCL "Accu"
       +
      STO "Accu"
      RCL "High"
      RCL "Incr"
      X<Y?
      GTO 01
      RCL "Win"
```

```
RCL "Accru"
  X<Y?
GTO 04
 LBL 05
 CLST
 STO "Accru"
 STO "Iter"
  RCL "Elem"
   1
  +
 STO "Elem"
  RCL "High"
 X=Y
 X≤Y?
 GTO 02
  LBL 04
 STO "Win"
 RCL "Elem"
  STO "Wt"
  GTO 05
· LBL 03
   RCL "Wt"
   RCL IND ST X
     STOP
     .END.
```

of 83 any natural elements (in complete analysis)

83 nat. elements 55 minutes to run

~~Tantalum 181~~ Thulium 169 is the last "whole more" stable ~~#73~~ #69 on the Periodic Table.

Upon careful review of the data and limiting the test to only naturally occurring isotopes, element #69, 168.934245000 - Tm^{169}. ~~Ta 180~~

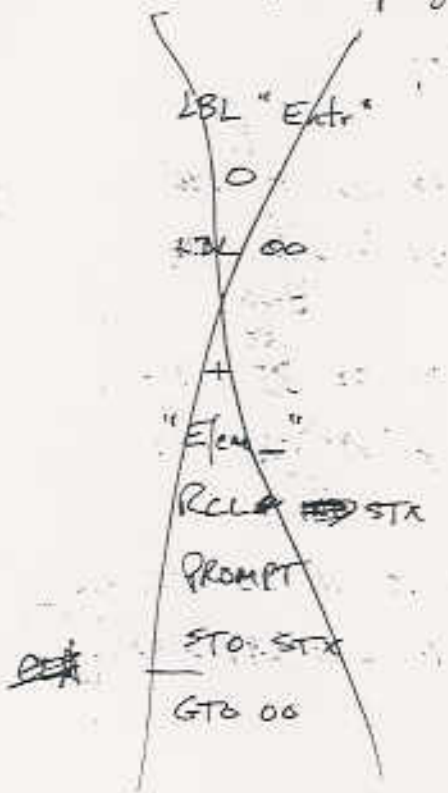
Thulium

Tm^{169} Makes up 100% of the naturally occuring element of 16 isotopes! - rare earth metal.

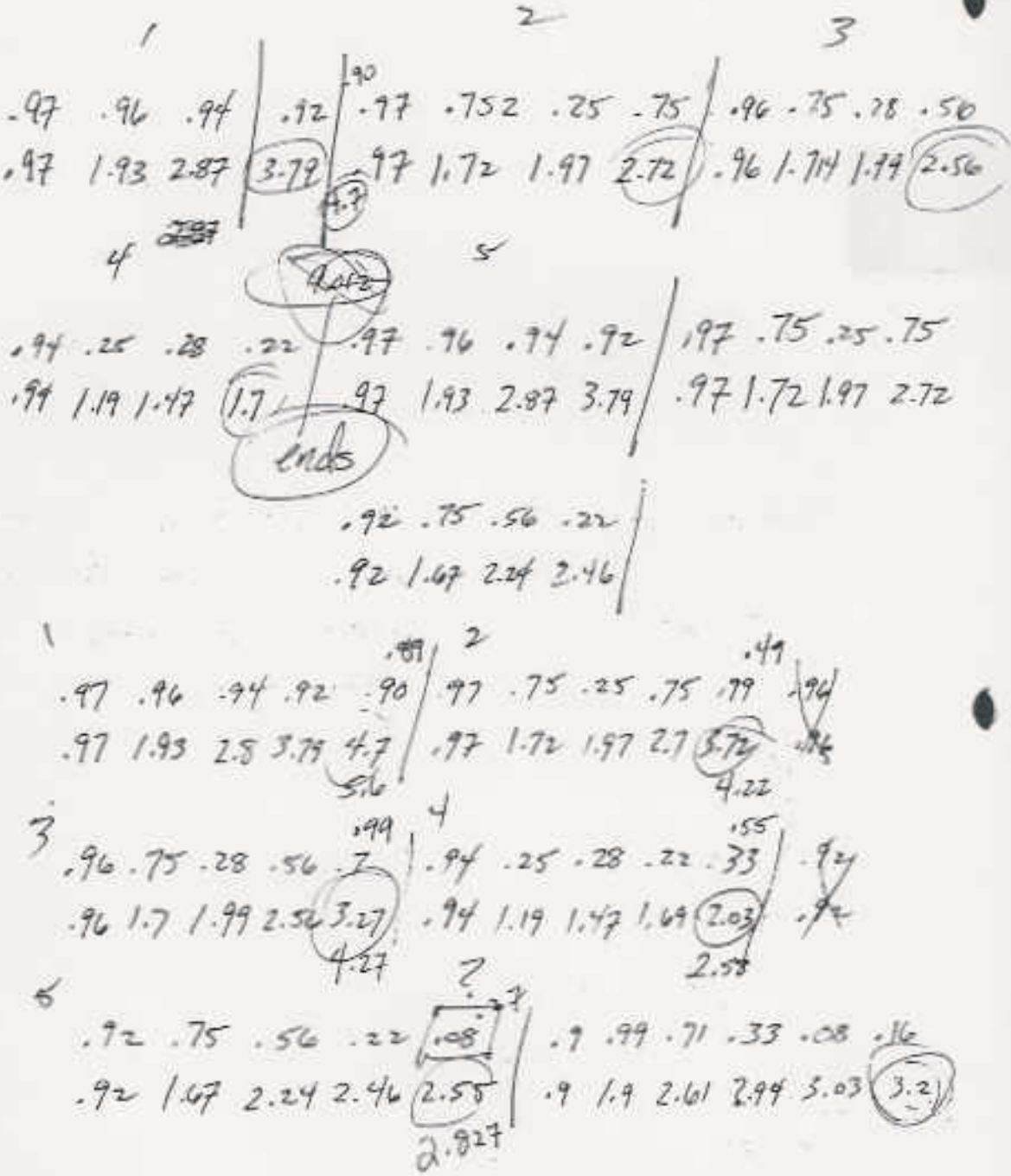
Reallocate registers in the sequences.
So not to interfere with the weight standard program.

00 Remember - this standard program is only single use.

Data load program.



3/14 - Upon installation of new batteries - reload 105 natural isotopes for various further manipulation.


1
2
3
.97 .96 .94 .92 .90 .97 .752 .25 .75 .96 .75 .28 .50
.97 1.93 2.87 3.79 4.7 .97 1.72 1.97 2.72 .96 1.714 1.99 2.56
4
5
.94 .25 .28 .22 .97 .96 .94 .92 .97 .75 .25 .75
.94 1.19 1.47 1.7 .97 1.93 2.87 3.79 .97 1.72 1.97 2.72
ends
.92 .75 .56 .22
.92 1.67 2.24 2.46
1
.89
.90
.97 .96 .94 .92 .90 .97 .75 .25 .75 .79 .94
.97 1.93 2.8 3.75 4.7 .97 1.72 1.97 2.7 3.72 .84
5.6
4.22
3
.99
4
.55
.96 .75 .28 .56 .7 .94 .25 .28 .22 .33 .92
.96 1.7 1.99 2.56 3.27 .94 1.19 1.47 1.69 2.03 .92
4.27
2.58
6
7
.92 .75 .56 .22 .08 .9 .99 .71 .33 .08 .16
.92 1.67 2.24 2.46 2.55 .9 1.9 2.61 2.94 3.03 3.2
2.827

Restructure with indirect addressing to accomodate all the stable nuclides.

Data entry:

```
LBL "STND"
  "H1"
  PROMPT
  STO "H1"
  "H2"
  PROMPT
  STO "H2"
  "He3"
  PROMPT
  STO "He3"
  "He4"
  PROMPT
  STO "He4"
  "Li6"
  PROMPT
  STO "Li6"
  "Li7"
  PROMPT
  STO "Li7"
  PROMPT
    :  Be9
    :   :
```

Two tables of data


```
RCL "He3"
RCL "H1"
x≠y?
x<y
÷
x≠1?    same element
IP      truncate integer
```

Running all 185 natural isotopes yields:

Lanthanide series 60 -
Neodymium 145 to be the last
whole number standard

△

P.S. 1st iteration

The final winner is Gadolinium 157
@ Old Table + 0.075975

(it took 5-6 iterations)

104

Clear all memory
6977 bytes

4673 bytes after
creating a 284×1 Matrix

"1"-"284" Alpha Variables

LBL 00

FP

Lanthanide Series 60

wt - 144.912538

Eform - 194.000

Win - 95.77740

Neodymium 145

To contain Plutonium 238 7 isotopes - nuclides - 142-146
148, 150

average the 285 for 286th Abund. 0.830
value.

0

RCL IND STX

ENTER

LBL 1

+

RCL IND STX

STO "WM"

X⇄Y

Turn "Reg" into (284x1) matrix
(284x1) matrix "Isotope"

```
LBL "Stnd"
 284
STO "High"
 0
STO "Accru"
STO "Frac"
STO "Low"
 1000
STO "Win"
CLST
 1
STO "Incr"
STO "Elem"
LBL 02
1
INDEX "Isotope"
RCL IND STK STO
RCL
GTO 00
LBL 01
RCL "High"
RCL "Elem"
X≥?
GTO 03
RCL "Incr"
 1
+
STO "Incr"
GTO 02
LBL 00
RCL "Elem"
1
INDEX "Isotope"
RCL IND STK STO
RCL
X⇄Y
CLX
2
R↓
X=Y?
GTO 01
X>Y?
X⇄Y
÷
FP
STO "Frac"
RCL "Accru"
+
STO "Accru"
RCL "High"
RCL "Incr"
X<Y?
GTO 01
RCL "Win"
RCL "Accru"
X<Y?
GTO 04
LBL 05
CLST
STO "Accru"
STO "Incr"
RCL "Elem"
 1
+
STO "Elem"
RCL "High"
X⇄Y
X≤Y?
GTO 02
LBL 04
STO "Win"
RCL "Elem"
STO "Wt"
GTO 05
LBL 03
RCL "Wt"
1
INDEX "Isotope"
RCL IND STK STO
RCL
"Isotope"
AXEL STZ
AVIEW
```

ProbeScope Windows Application V4.1 1996
Monday the 19th, March 19101 at 08:32:56
Filename: A:\MYNAME
File Commentary: Entered DVYD MRAL #31244154.

Timebase:	10ms/div 1kSa/s
Trigger:	+INTERN\SINGLE
Input Coupling:	AC
Input Voltage:	250mV/div

Delta Time in HERO.02632 KHz

Interpreter: English to Hebrew consonants and
connector to numeric meaning in Hebrew.

For example:

enter David = 3160 Hex character codes.

4 10 4 = 2# or 1#

- remove vowels from spelling -
- return (sound) a word or name -
- return Hebrew spelling -

utilize ATOX feature
have a vowel subroutine return "Ø".

(ASCII) STORE "DAVID" in "NAME"
RCL "NAME"
ATOX

Rcl (Array (Matrix) of vowel values) in x reg
compare - if they equal a vowel, return Ø

(360)

Sum these values

Convert the values to
Hebrew values.

31

32 Space

X

digit					
0 48		A=65 = ~~Aleph~~ Ø	Ø	97	Just evaluate consonants
1 49	1	B=66 = Beth	2	98	
2 50					
3 51	2	C=67 = Caph	20	99	~~ATOX~~ will give the character
4 52	3	D=68 = Daleth	4	100	code ⇒ Compare to a matrix
5 53		Yod E=69 = Ø ~~Aleph~~		101	and then when the number
6 54	4	F=70 = Fey	80	102	in the character code matrix is
7 55	5	G=71 = Gimel	3	103	found - compare to another matrix
8 56	6	H=72 = Heth	5	104	of Hebrew alpha letters - better
9 57		Heth I=73 = Ø Yod		105	~~learn~~ the Hebrew alpha-letters to
	7	J=74 = Jod	10	106	the character code - simpler.
	8	K=75 = Koph	100	107	
	9	L=76 = Lamed	30	108	
	10	M=77 = Mem	40	109	
	11	N=78 = Nun	50	110	
		Heela O=79 = Ø ~~Jaw~~		111	
	12	P=80 = Fey	80	112	
	13	Q=81 = Koph	100	113	
	14	R=82 = Resh	200	114	
	15	S=83 = Shin	300	115	
	16	T=84 = Tav	400	116	
		Aleph U=85 = Ø Jaw		117	
	17	V=86 = Vaw	6	118	
	18	W=87 = Vaw	6	119	
		X=88 = Ø ~~Koph~~		120	
	18	Y=89 = Jod	10	121	
	19	Z=90 = Zain	7	122	

X	Y
66	"Beth"
67	"Caph"

CAP LETTER #

X⇄Y

X=Y?

XEQ "HEBREW"

SMALL LETTER#

X⇄Y

X=Y?

XEQ "HEBREW"

First I'll establish these parameters as automatically growing matrices.

Example

Aleph ⇒

Symbol	Category	Character
oxen	Mother I	Moistness

Beth

Symbol	Category	Character	Celestial	~~day~~ time	Sephirotic assignment: organ	direction
house	double ~~[illegible]~~ 1	Wisdom/ignorance	Moon	1st	right eye	

He

Symbol	Category	character	Celestial	~~day~~ time	organ	direction	Sephirot
Window	simple 1	speaking	Aries/~~[illegible]~~	Nisan	right foot	high east	5 ~~[illegible]~~

Practicality dictates that ~~registers~~ labels 00-99 be utilized restricting to 22 x 4 = 88 fields (alpha) about each consonant. More like 19 x 5 = 95 alpha fields.
(19x4 but growable) ~~[illegible]~~ 76

Symbols	Technical	Space/time	Sephirot	(open)
symbol action (speaking)	single/double linguistic class	Celestial day/month organ direction	# name	

ASTO "Name" Name = "Tetpoid"
ATOX "68"

LBL "Daleth" 68
A XEQ "Daleth"

– Upper case & lower case letters –

A	65	a	97	0	48
B	66	b	98	1	49
⋮		⋮		⋮	
Z	90	z	122	9	57

ATOX
65
X⇄Y
ATOX
X=Y?

toggle flag for upper or lower case "Aleph" to begin a three consonant Hebrew word.

XEQ "Aleph"

LBL "Cabala"
LBL 01
ATOX 97 ENTER (lower case)
65
X⇄Y
X=Y?
XEQ "Aleph" RL, X=Y?, XEQ "Aleph"
66
X⇄Y
X=Y?
XEQ "Beth"

COST

0
0
68 ← "D"
68
65 ← "A"
65
68
68
97
97
68 ← Keep shuffling to bottom (X)

```
LBL "Cabala"
LBL 01
CLST
ATOX
65
X≷Y
X=Y?
XEQ "Aleph"
X≷Y
X=Y? yes  XEQ "Aleph"
XEQ "Aleph"
66
X≷Y
X=Y?
XEQ "Beth"
98
X≷Y
X=Y?
XEQ "Beth"
67
```

```
0
0
0
97 ← small a
97
65
65
97
97
97
97
97
```

```
0
0
0
118 ← small v
118
65
65
118
118
97
97
118
```

```
0
0
0
100 ← small d
100
65
65
100
100
97
97
100
100
"B" → 66
```

215 lines for all letters —
260 with digits - the digits had to be {bracketed} to become 'alpha' digits.

'Cabala' sends (goes) for a 'letter' program.
Primarily, a letter program is a display routine containing the philosophical symbols associated with the Hebrew letters.

'Gematria' identifies the Hebrew numeric value and sums up the name or word. The display is the name or word and its Hebrew total.

'Dabar' is a collection of the first three Consonant sounds of each word or name. If memory allows experimentation a dictionary will identify vocabulary and display definitions.

'Shamine' is a 22x4 matrix. Once the programs are set and running, symbols or quantity, or representative values can be manipulated in imaginative ways. Correlations sought - whatever.

~~outline~~ ~~back two leafs~~

"Cabala" letter display routines.

#4	Vau
00 {41-Byte Prgm}	00 {62-Byte Prgm}
01 LBL "{4}"	01 LBL "Vau"
02 "Hesed-mercy"	02 View "Vau"
03 AVIEW	03 PSE
04 PSE	04 ~~Nail to bind~~ [Nail-to-bind] ~~"View" Nail~~"
05 "Evil"	05 AVIEW
06 AVIEW	06 PSE
07 PSE	07 "Taurus - Iyar"
08 "Daleth, Nun"	08 AVIEW
09 AVIEW	09 PSE
10 ~~PSE~~ ~~ARCL "Name"~~ CLA	10 "Thinking/re-kid"
11 END	11 AVIEW
	12 PSE
	13 ~~STOP~~ ~~ARCL "Name"~~ CLA
	14 END

Gematria

Implement counting-summation registers for 1) Hebrew total (w/spelling & val) 2) English (A=1; Z-9) breakdown (3 stages) 3) ~~ASCII~~ (+) summation & breakdown.

0
STO WRD1
STO WRD2

STORE these numbers (letters) in an array (matrix) (1X18)

A B C D E F G H I J K L M N O P Q R
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

LBL "Cabala"

OLA

"Name or Word"

only capitals first 5 letters! PROMPT

ASTO "Name" "David"

LBL 01

CLST ~~ASTO "Name"~~ ← place in the program at end. :- direct

D=68 ATO X "avid"

ASTO "Name" "avid"

68 48

X⇄Y

X=Y?

Jump to Delete

XEQ "Delete" XEQ "E03" ← (Aquires new alpha data for display of symbols)

⋮

XEQ "Join"

31 ← "Ø"

X≠Y?

GTO 01

"Done"

AVIEW

STOP

END

Display

Make "Name" a 1x19 Matrix

(CLST) (CLA)

"ABC"

– PROMPT –

INDEX "NAME"

19

ENTER

1

ENTER

LBL 02

STOIJ

ATOX X:65 X:66

STOEL

I+

R↓

x<>y

1

+

x≠y ?

GTO 03

[RCLEL routine] between lines 20 & 21

48

–

LBL 03

x<>y

1

x<>y

GTO 02

@ end of program following GTO 01

[RCLEL routine]

– GTO 03 –

INDEX "Name"

1

ENTER

STOIJ

LBL 01

RCLEL

48

–

I+ after XEQ "Zin"

31

x≠y ?

GTO 01

x=y ?

GTO 04

[LBL 03]

LBL 04

"Done"

AVIEW

STOP

(288) END

1498 bytes available.

Gematra — the program "Gematra" will be addressed from the independent programs for each Hebrew letter. This is a single gosub. GTO "Gematra" and then it will return to call the current ~~ASCII~~ # and then return to "Cabala". The end of the "Cabala" program will display results of the Gematra.

Objectives - sum up Hebrew letters (consonants)
- Removed send command from Aleph.
- Sum up the ~~ASCII~~ amounts of each word.
- Break down the English 1=A, 2=B etc.

RCL "letter #" before sending to "Gematra"

~~A second 1x19 matrix will hold the English numerology. Easier — store the English # in the Hebrew letter program. Wont work — Vaw = Vaw (v) (w) and some English letters are non-existent in Hebrew. A second matrix but call the Hebrew letter values to the stack before "Gematra".~~

(1x9)
(3x9)?

"English" 1x26 matrix. for use in "Gematria"

→	1	2	3	4	5	6	7	8	9
.	65/97 A	66/98 B	67/99 C	68/100 D	69/101 E	70/102 F	71/103 G	72/104 H	73/105 I
..	74/106 J	75/107 K	76/108 L	77/109 M	78/110 N	79/111 O	80/112 P	81/113 Q	82/114 R
∴	83/115 S	84/116 T	85/117 U	86/118 V	87/119 W	88/120 X	89/121 Y	90/122 Z	32 SP

31

65/97 ⇒ 1,1 66/98 ⇒ 1,2 so on ... ⇒

74/106 75/107

83/115 84/116

Alpha register — "ACE"

. Atbx produces 65 in X reg. All the while, the name is stored in the "Name" matrix. So...

for example: BILL GATES 3 (III) = 666

1415 bytes → "Gematria" receives the ASCII value in y: and the Hebrew letter value in x: for summation, etc.
before

Summation
(Program)

Name1	Asc1	Name2	Asc2
"Word1"	Heb1	"Word2"	Heb2

```
LBL 04
STO+ Heb2
x⇄y
STO+ Asc2
```

```
STO+ "Heb1"
x⇄y
STO+ "Asc1"
end
```

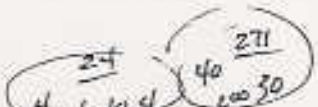
```
x⇄y
31
x=y?
GTO (end display)
CLX
+
32
x=y?
SF 00
FS?
GTO 04
```

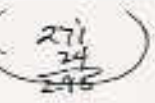
00 CF at beginning of program

at entry

```
x⇄y
31
x=y
GTO (end display)
```

```
SF 00  CLX
x⇄y    +
FS 00?
GTO 04
```



Results 

David_Mral 24 271 = 295 Hebrew

391	396	787	ASCII

Gamatria | David_Merrill 488 + 727 = 1,215 ASCII

```
LBL "Gamatria"
X⇄Y
31
X=Y?
GTO 05
CLX
+
32
X=Y?
SF 00
CLX
+
X⇄Y
FS? 00
GTO 04
STO+ "Heb1"
X⇄Y
STO+ "ASC1"
GTO 05
LBL 04
STO+ "Heb2"
X⇄Y
STO+ "ASC2"
CLX
LBL 05
END
```

```
Display results - "Cabala" ~300
LBL 04
"Hebrew sum is
+ "_"
ARCL "Heb1"
+ "_F ASCII sum
+ "_is_"
ARCL "ASC1"
AVIEW
PSE (x4)
FC? 00
GTO 05
"Second part"
+ "_sums to F"
ARCL "Heb2"
+ "_and in ASCII
RCL "ASC2"
32
-
STO "ASC2"
ARCL "ASC2"
AVIEW
PSE (x4)
LBL 05
END
```

```
RCL "Heb1"
RCL "Heb2"
+
RCL "ASC1"
RCL "ASC2"
+
"Hebrew ~~summation~~"
+ "_total is_"
ARCL ST Y
~~AVIEW~~
~~PSE (x4)~~
+ "ASCII total is_"
ARCL ST X
AVIEW
PSE (x4)
```

Matrix (1X5) "Sum"

CLA
ALEPH — ASP — A

Total sums — B

"281" — C

1,1; 1,2; 1,3 ... — D

2 8 1 — E

It — F

2+8+1+0+0 — G

" — H

1,1; 1,2; 1,3; ... — I

1+1+0+0+0 — J

2 — K

L M N O P Q R S T U V W X Y

Backwards!!

I wish to list the Hebrew alphabet - using English letters -

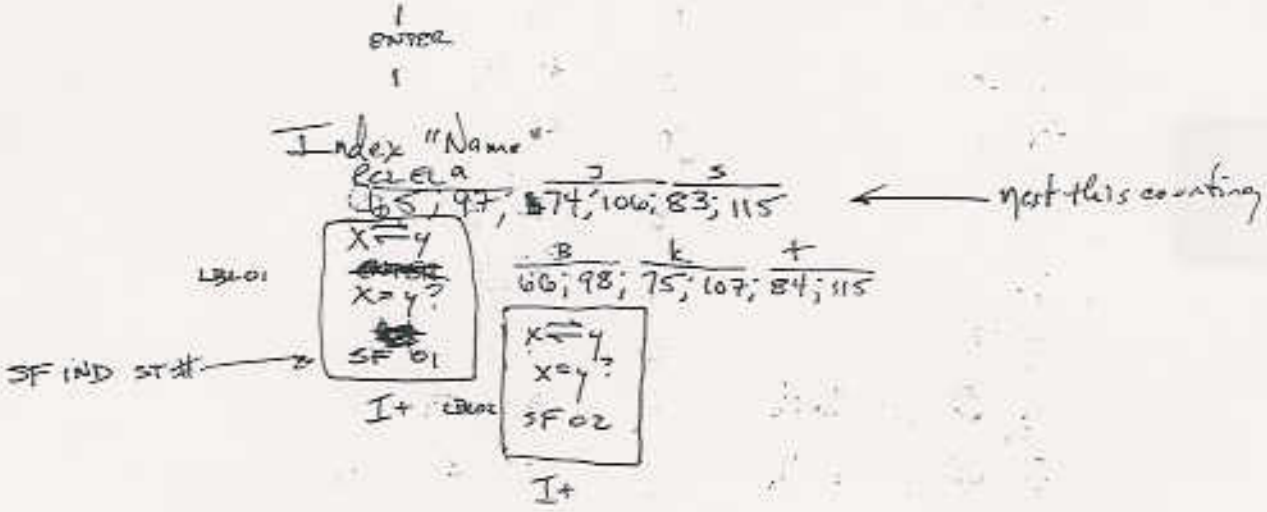
"O" is for Chet

"E" is for Teth

"U" is for Samech

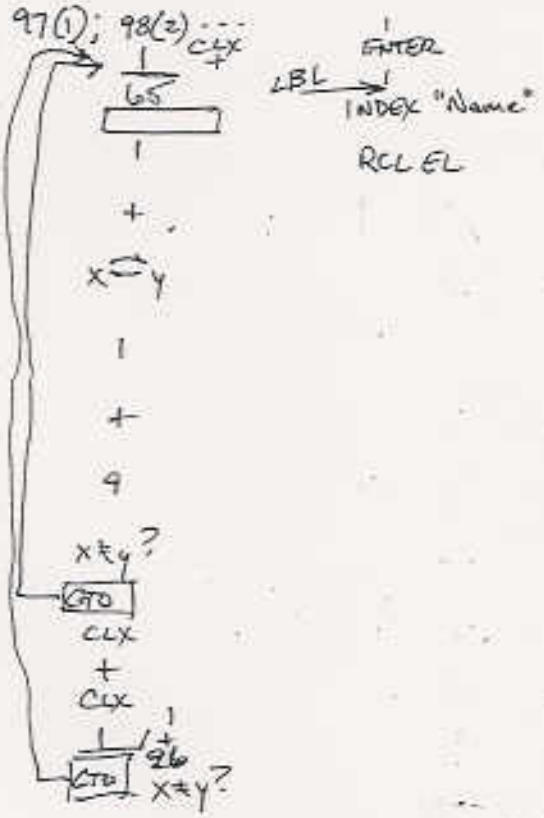
"I" is for Ayin

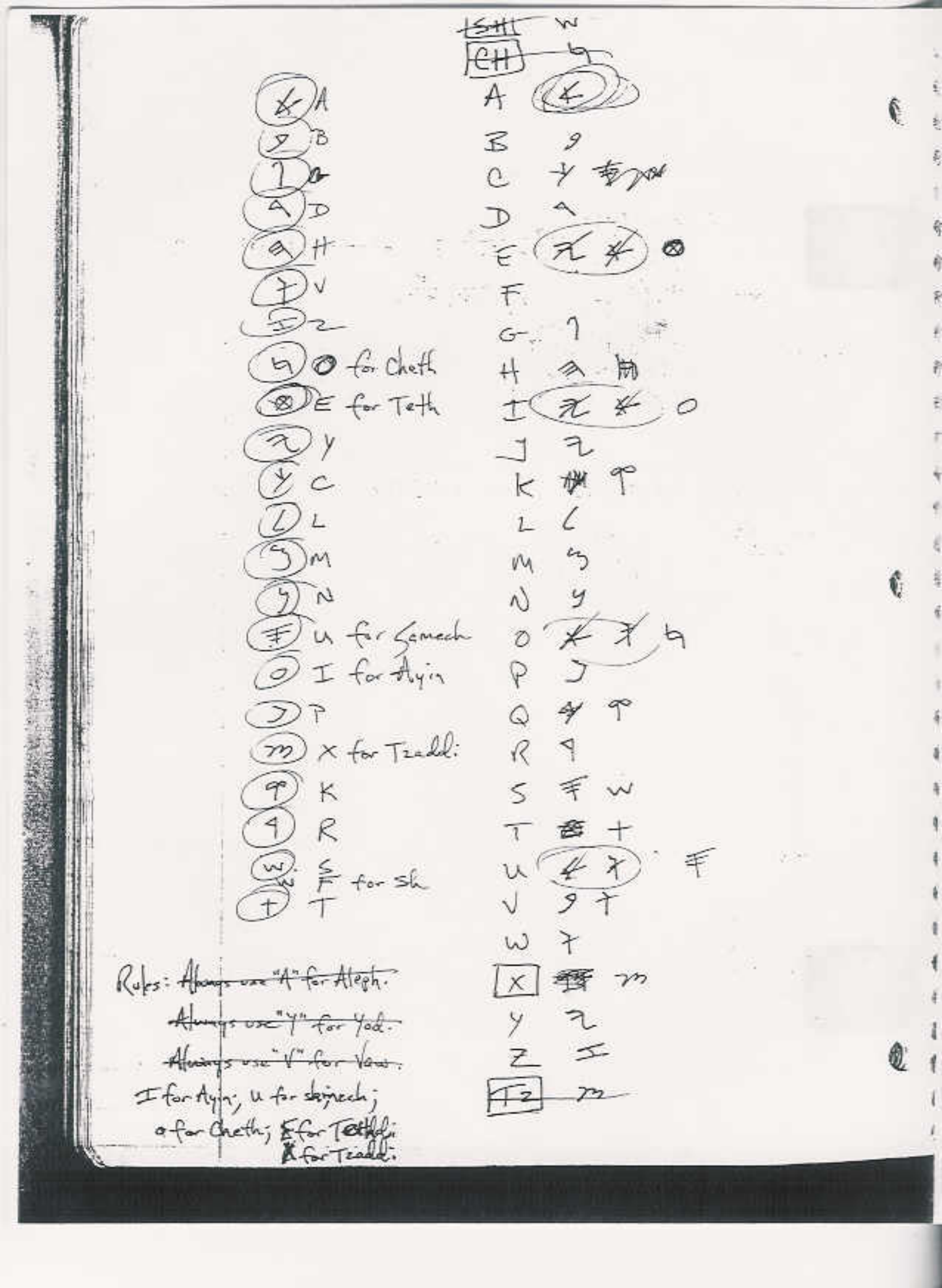
"X" is for Tzaddi



65(1) ; 66(2); ··· 73(9); 74(1); 75(2) ··· 82(9); 83(1) ···

97(1); 98(2) ···




~~SH~~ W
CH
A
B
C
D
E
F
G
H
V
Z
for Cheth
for Teth
Y
C
L
M
N
U for Samech
I for Ayin
P
X for Tzadi:
K
R
S for Sh
T
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
Rules: ~~Always use "A" for Aleph.~~
~~Always use "Y" for Yod.~~
~~Always use "V" for Vav.~~
I for Ayin; U for Shinech;
O for Cheth; ~~E for Teth~~;
~~X~~ for Tzadi.

O cheth
C Teth
U Samech
I Ayin
X Tzaddi

After streamlining out the Sepher Yesirah four and adopting some rules to include the entire Hebrew alphabet in numerical translation, I can examine hebrew words.

~~Agreement~~

[illegible]	chozeh Seer, vision	- 2	Isaiah 28:15
[illegible]	chazuth vision	- 7	Isaiah 28:15
[illegible]	upright things Meysharim	- 6	Daniel 11:6
	[illegible] ~~manda~~	- 1	
[illegible]	Amen	- 1	
[illegible]	Shalom Amends/peace	- 1	
[illegible]	Ammon a fellow countryman	- 4 (12)	
[illegible]	Amnon tutelage, upbringing	= 12 (3)	
[illegible]	Beyn between, understand	- 8	
[illegible]	Amorite mountaineer	- 8	
[illegible]	Amram exalted people	- 5	
[illegible]	Anakim giants	- 9	
[illegible]	Cabal ~~Secret meeting~~ agreement	- 6	
[illegible]	Cabala Secret meeting	- (11) 2	
[illegible]	owth agreement/Mark	- (11) 2	

Ø 48
/ [illegible] 31

Numero-linguistic standing wave shifts power (gevurah) into judgment.

Perfect Prayer — in numerics

stamina			gates →		words →		
[illegible]	A	65					
[illegible]	B	66					
[illegible]	G	71					
[illegible]	D	68					
[illegible]	H	72					
([illegible]	F	70)					
[illegible]	V	86					
[illegible]	Z	90					
[illegible]	O	79					
[illegible]	E	69					
([illegible]	J	74)					
[illegible]	Y	89					
[illegible]	C	67					
[illegible]	L	76					
[illegible]	M	77					
[illegible]	N	78					
[illegible]	U	85					
[illegible]	I	73					
([illegible]	Q	81)					
[illegible]	P	80					
[illegible]	X	88					
[illegible]	K	75					
[illegible]	R	82					
W	S	83					
+	T	84					

(22) (231) (10,164)

"tikkun" = restoration.

(reduce "Name" + 1×3 matrix)

discard the Sepher Yetzirah as Jewish philosophy

10sec/3 [illegible]

+ discard ASCII info.

Exercising the interface

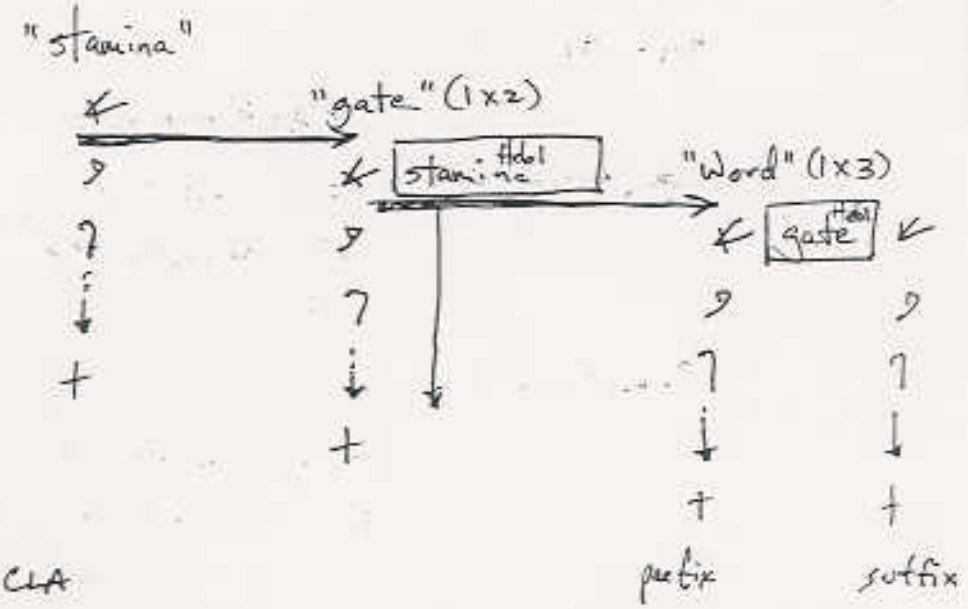
stamina	gates	words
+	+	+
x ⇌ y	x ⇌ y	x ⇌ y
x = y?	x = y?	x = y?
GTO "Aleph"	GTO "Aleph"	GTO "Aleph"
CLX + 2	nest "stamina"	nest gates
x ⇌ y	run "Cabala"	run "Cabala"
x = y?	CLX + 2	CLX + 2
GTO "Beth"	x ⇌ y	x ⇌ y
CLX + 3	x = y?	x = y?
⋮	GTO "Beth"	GTO "Beth"
↓	↓	↓

Assembling the ultimate expression.

stamina	gate	word
Aleph = 1		
	Aleph, Aleph = 2	
		Aleph, Aleph, Aleph = 3
		Beth, Aleph, Aleph = 4
		Gimel, Aleph, Aleph = 5
		⋮
	Beth, Aleph = 3	
		Aleph, Beth, Aleph = 4
		Beth, Beth, Aleph = 5
		Gimel, Beth, Aleph = 6
		⋮
	Gimel, Aleph = 4	Aleph, Gimel, Aleph = 5
		Beth, Gimel, Aleph = 6
		Gimel, Gimel, Aleph = 7
	⋮	⋮
Beth = 2		
	Aleph, Beth = 3	
		Aleph, Aleph, Beth = 4
		Beth, Aleph, Beth = 5
		Gimel, Aleph, Beth = 6
		⋮
	Beth, Beth	
		Aleph, Beth, Beth = 5
		Beth, Beth, Beth = 6
		⋮

"Matrix"

Replace "Nama" in "Cabala" with a cascading 1x3 matrix



CLA

CLST

~~XQ~~

STO "HEB1"

STO "HEB2"

CF 00
Fachian Var.
XEQ "WORD"

indirect variable "Word", "gate" or "Stamina"

Fashion variable initial setup ~40 lines

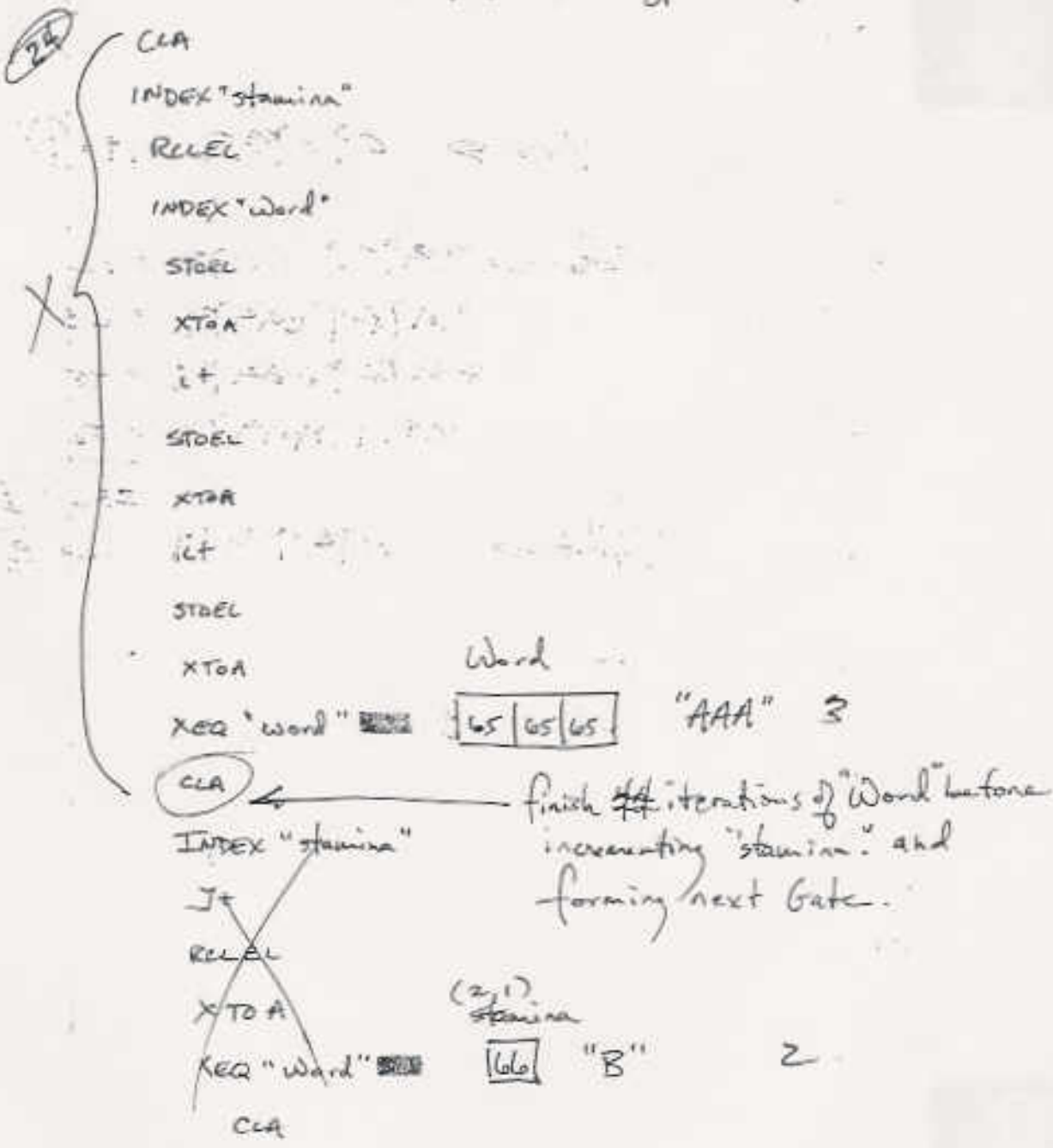
```
LBL "Matrix"
 CLA
 CLST
 STO "Hdb1"
 CF 00
Index "stamina"
  1
 ENTER
 LBL 01
 STO IJ
 RCLEL
 XTOA
 XEQ "Word"  [65] "A" 1
  CLA
INDEX "stamina"
RCLEL
INDEX "Gate"
 STOEL
 XTOA
  1+
STOEL
XTOA
XEQ "Word"  [65][65] "AA" 2
```

Jesus ⇒ 07W737Z = 19 = (1)

Father ⇒ 2737Z7Z2A77Z = 72
2737K7Z9A77Z = 63
K737K7K3A77Z = 45
3377737A77Z = 52
232 = (7)

Holy Spirit ⇒ W7A7479Y = 12 = (3)

instead load current "Gate" @ "Name" (1,2)$(1,3)



22 iterations of "Word".

(B)

CLA

CLST

1

1 2

2 LBL 03

STO "Index"

LBL 02

R↓

3 1

INDEX "stamina"

STO I]

RCLEL

R↓

X⇄Y

R↓

INDEX "Word"

STO I]

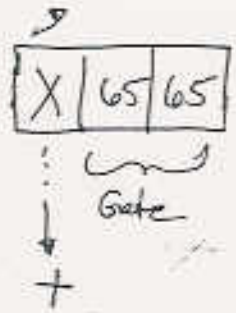
R↓

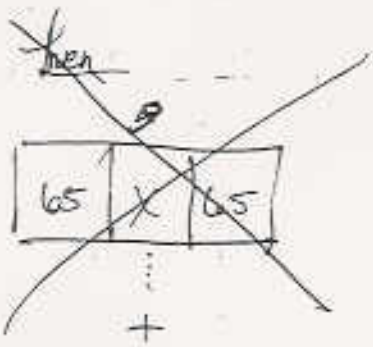
R↓

STOEL

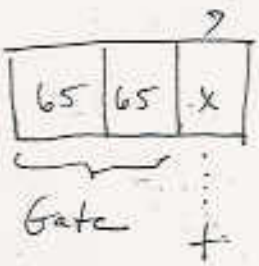
XTOA

I+





Then



44 ~~to~~ iterations each letter 1,452.

Well done but this leaves a lot of repetition to cover all words

CLA
1
RCL "Index"
1
+
STO "Index"
23
(3)→ X≠Y?
GTO 02
GTO 03 (=)
GTO 04 (@ line 24)

Prepare stack for 2nd iteration for "Word" 1X3:

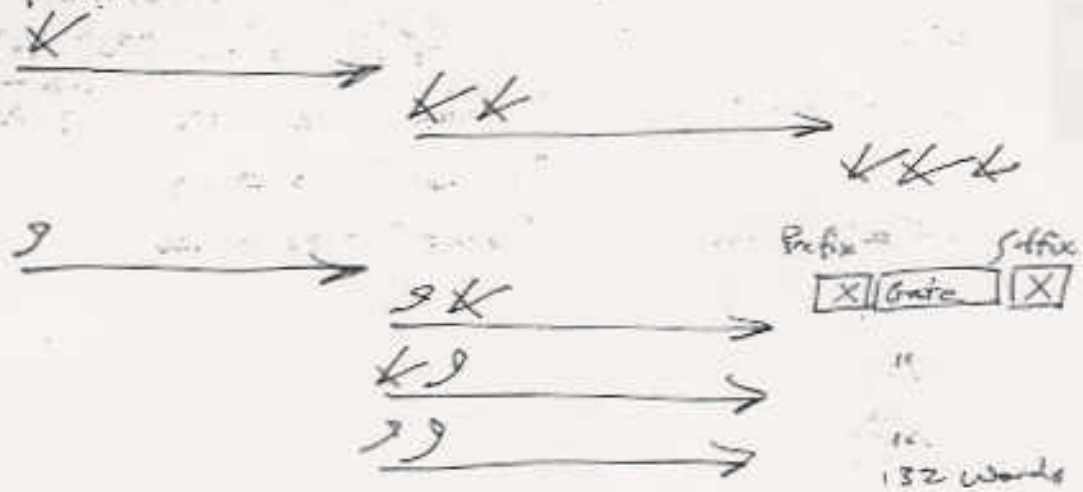
▷	= 2
▷ ✕	= 3
▷ ✕ ✕	= 4
✕ ▷ ✕	= 4
✕ ✕ ▷	= 4
▷ ▷ ✕	= 5
✕ ▷ ▷	= 5
▷ ▷ ▷	= 6

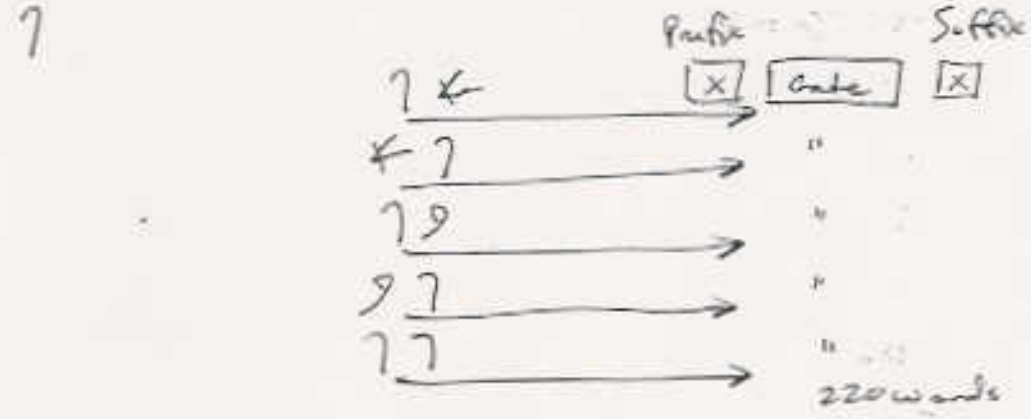
✕ ▷	= 3
▷ ▷	= 4

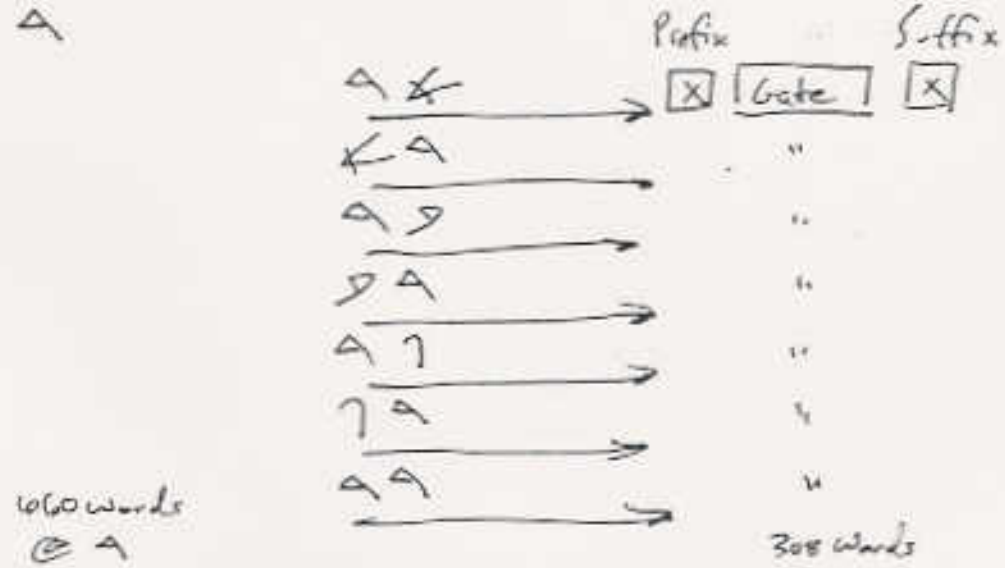
Cascading arabic combinations already display 10 combinations at 9.

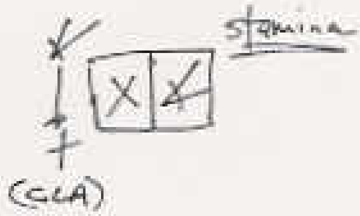
Doubles the "Gate".

"Stamina" "Gate" "Word"









3/14 "Dimension error @ Tav, Ceph, Alph due to $\frac{23}{1}$ y x reducing "23"'s to "22"

"Index" = 22 @ 106

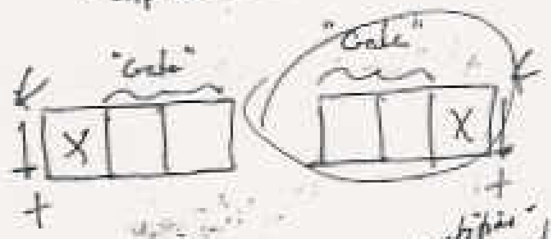
"Number" = 12 @ 115

"Carat" = 22 @ 126

```
LBL 04
RCL "Number"
 1
INDEX "stamin"
 STOIJ
 RCLEL
 X<>Y
  1
 INDEX "Gate"
  STOIJ
   R↓
   R↓
 STOEL
 XTOA
 I+
 RCLEL
 XTOA
 XEQ "Word
  CLA
```

load "Gate" into "Name" to scan alphabet.

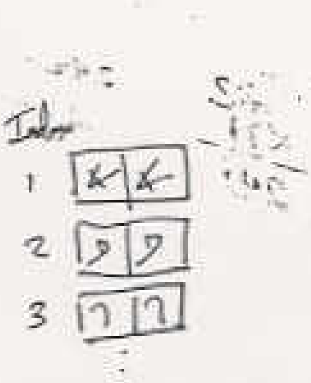
"Matrix"



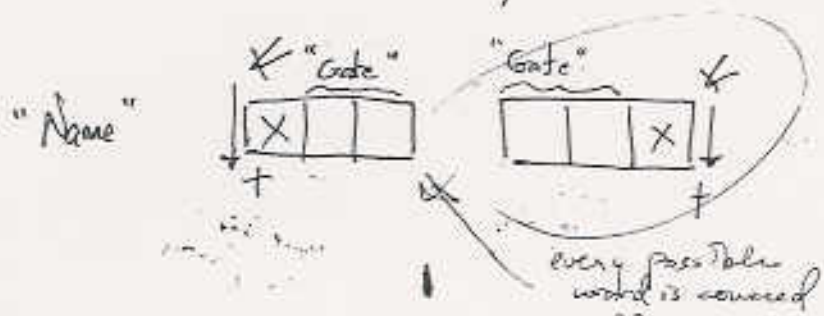
3/8/01

```
LBL "Matrix"
CLA
CLST
STO "Hel1"
STO "Hel2"
CF 00
1
1
STO "Count"
STO "Number"
LBL 01
RL
1
INDEX "Stamina"
STO IJ
RCLEL
XTOA
XEQ "Word"
CLA
RCL "Count"
Index "stamina"
STOIJ
RCLEL        XFY
              1
INDEX "Gate"
STOIJ        RL RL
STOEL          RL
XTOA
I+
STOEL
XTOA
XEQ "Word"
CLA
```

```
INDEX "Gate"
STO IJ
RCLEL
I+
RCLEL
1            RL
3            RL
INDEX "Name"
STOIJ
RL
RL
STOEL
I-           I+
RL
STOEL
CLST
CLA
```



Sending "Name" to "Cabala" by loading into Alpha register.



every pass Table word is counted

23
22
5083 words

```
1
1
STO "INDEX"
LBL 02          LBL 03
                RL
  X⇄Y           X⇄Y
  1             RT
  X⇄Y
INDEX "stamina"
STO IJ
RCL EL
X⇄Y
.1 , 3

INDEX "Name"
STO IJ
RL
RL
STO EL
XTOA
I+   I-
     I-
RCLEL
XTOA

I+ I+
RCLEL
XTOA          I+ RCLEL
              XTOA
XEQ "Word"
CLA
1
RCL "Index"
1
+
STO "Index"
X⇄Y
23 22
X≠Y?
GTO 02        GTO 03
              CLA
              RCL "Count"
              1
              -
              STO "Count"
              23 22
              X⇄Y
              X≠Y?
              GTO 01
RCL "Number"
1
+
STO "Number"
X⇄Y
.12
X≠Y?
GTO 04
RCL "Number"
1
STO "Number"
484
11
5324
```

* shorten "Name" to a 1x4 Matrix to reduce "Matrix" run time. Otherwise 1x18 —

"Cabala" example DVYD_MRAL (Enter)

daleth, Vaw, Yod, daleth _ Mem, Resh, Aleph, Lamed

4, 6, 10, 4 (24) 40, 200, 1, 30 (271)

00 {517 - Byte Prgm}

```
01 LBL "Cabala"
   CLA
   CLST
   0
   STO "Heb1"
   STO "Heb2"
   CF 00
   "Enter name or -"
   ↑ word
   ↑ "L_F"
   PROMPT
   LBL "Word"        <- entry for "Matrix" to run "Word" values.
   ↑ "□"
   INDEX "Name"
(1x18) (19)5 - [1x4] *
   ENTER
   1
   ENTER
   LBL 02
   STOIJ
   +
   ATOX
   STOEL
   It        271 + 24 = 295
   Rb
   x≠y?      2 9 +5 = 16     1 +6 = (7)
   GTO 03
   INDEX "Name"
   1
   ENTER
   STOIJ
   LBL 01
   RCLEL
   32
   X⇄Y
   X=Y?
   XEQ "space"
   65
   X⇄Y
   X=Y?
   XEQ "Aleph"
   66
   X⇄Y
   X=Y?
   XEQ "Beth"
   V
```

(134)

```
90
X≠Y
X=Y?
XEQ "Zain"
I+
31
X≠Y?
GTO 01
X=Y?
GTO 04
LBL 03
1
X≠Y
GTO 02
LBL 04
"Hebrew sum is"
"_"
ARCL "Heb1"
AVIEW
PSE
"LF"
FC? 00
GTO 05
"Second part is"
ARCL "Heb2"
"LF"
AVIEW
PSE
PSE
LBL 05
RCL "Heb1"
RCL "Heb2"
+
"Hebrew total_
ARCL ST X
"LF"
AVIEW
PSE
PSE
0
STO "Heb1"
STO "Heb2"
RTN
171 END
```

Variable Registers

Counters		Matrices	Numeric stores
Index	{0}	stamina (22x1)	Heb1
Counter	0 value for space.	gate (1x2)	Heb2
Number		gene (1x4)	
		regs (25x1)	

"Cabala" Subroutines

Sends "Ø" letter value for space to "Gamatra".

{41-Byte Prgm}
LBL "space"
"Next word is "
AVIEW
PSE
CLA
RCL "Ø" = Ø
GTO "Gamatra"
END

Variable register "Ø"
Holds the value Ø.
Variable register "Aleph" (typical)

{34-Byte Prgm}
LBL "Aleph"
VIEW "Aleph"
PSE
CLA
RCL "Aleph"
GTO "Gamatra"
END

'Gamatra' sums words of

{50-Byte Prgm}
LBL "Gamatra"
X≠Y
31
X=Y?
GTO 05
CLX
+
32
X=Y
SF ØØ
CLX
+
X≠Y
FS? ØØ
GTO 04
STO+ "Heb1"
GTO 05
LBL 04
STO+ "Heb2"
LBL 05
CLX
END

(1)

"Matrix"

{514-Byte Prgm}

```
LBL "Matrix"
CLA
CLST
STO "Hdl"
STO "Hdl2"
CF 6 8
1
1
STO "Count"
STO "Number"
LBL 01
RL
1
INDEX "stamina"
STO IJ
RCLEL
XTOA
XEQ "word"
CLA
RCL "Count"
1
INDEX "stamina"
STO IJ
RCLEL
X≠Y
1
INDEX "Gate"
STO IJ
RL
RL
STOEL
XTOA
IT
STOEL
XTOA
XEQ "Word"
CLA
LBL 04
RCL "Number"
1
INDEX "stamina"
STO IJ
RCLEL
X≠Y
1
INDEX "Gate"
STO IJ
RL
```

(2)

```
R↓                      I-
STOEL                   R↓
XTOA                    STOEL
I+                      CLST
RCLEL                   CLA
XTOA                    1
XEQ "Word"              1
CLA                     STO "Index"
1                       LBL 02
1                       X⇄Y
INDEX "Gate"            1
STO 17                  INDEX "stamina"
RCLEL                   STO 17
I+                      RCLEL
RCLEL                   X⇄Y
1                       1
3                       INDEX "Name"
INDEX "Name"            STO 17
STO 17                  R↓
R↓                      R↓
R↓                      STOEL
STOEL                   XTOA
```

(3)

```
I+                  RCLEL
RCLEL               R↓
XTOA                R↓
I+                  INDEX "Name"
RCLEL               STO17
XTOA                R↓
XEQ "Word"          R↓
CLA                 STOEL
1                   I+
RCL "Index"         R↓
1                   STOEL
+                   CLST
STO "Index"         CLA
23                  1
X≠Y?                1
GTO 02              1
CLA                 STO "Index"
1                   LBL 03
1                   R↓
INDEX "Gate"        X⇄Y
STO17               INDEX "stamina"
RCLEL               STO17
I+                  RCLEL
```

eliminated 50
has no repetitive,
5,324 total words.
5,082 words
5,321 Gates

(4)

```
X≠y
 3
INDEX "Name"
STO 17
R↓
R↓
STOEL
I-
I-
RCLEL
XTOA
I+
RCLEL
XTOA
I+
RCLEL
XTOA
XEQ "Word"
CLA
 1
RCL "Index"
 1
 +
STO "Index"
 22
X≠y?
GTO 03
CLA
RCL "Number"
 1
 +
STO "Number"
 12
X≠y
X≠y?
GTO 04
RCL "Number"
 1
STO "Number"
RCL "Count"
 1
 +
STO "Count"
 22
X≠y
X≠y?
GTO 01
END
```

Collapsing the probability curve.

Reformulated standard selection. Selecting Neodymium 145 as a preferable standard to Carbon 12 was only the first iteration. 50 more lines adjust the table accordingly to 145/145 = 1 and rerun the program. The program ends when an element is selected twice in a row - it could take months at 14 hours per cycle. The program was (began) executed 3/17/01 @ 7:30 pm. Testing on the first 15 elements produced C^{13} the whole(ot) standard. 10 produced Hydrogen'.

```
LBL "Std"
10      (C^12) LastWinner
STO "LWin"
280
STO "High
LBL 07
0
STO "Accum"
STO "Free"
STO "Low"
```

```
1000
STO "Win"
CLST
1
STO "Elem"
STO "Incr"
LBL 02
1
INDEX "Data"
STO 17
RCL[illegible]
GTO 00
LBL 01
RCL "High"
RCL "Elem"
X≥y?
GTO 10
RCL "Incr"
(
+
STO "Incr"
GTO 02
```

RCL "L"Win"

RCL "Wf"

STO "LWin"

X=y?

GTO 03

1

INDEX "Isotope"

STO I

RCL EL

FP

STO "Frac"

1

ENTER

STO "Count"

RCL "Frac"

1

—

ABS

0.5

X>y?

GTO 08

1

ENTER

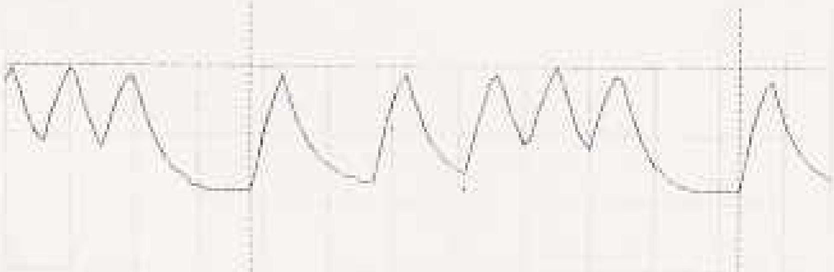
LBL 06
INDEX "Isotope"
STO J
RCL 02
RCL "Frac"
−
STO 02
RCL "Count"
1
+
STO "Count"
RCL "High"
x<y?
GTO 07
X⇄Y
1
GTO 06
(LBL 08)

LBL 08
1
ENTER
LBL 09
INDEX "Isotope"
STO J
RCL 02
RCL "Frac"
1
+
STO 02
RCL "Count"
1
+
STO "Count"
RCL "High"
x<y?
GTO 07
X⇄Y
1
GTO 09

LBL 07
RCL "Wt"
1
INDEX "Isotope"
STO 17
RCL EL
"Isotope_"
ARCL ST Z
AVIEW
LBL 11
BEEP
PSE
?
GTO 11
.END.

ProbeScope Windows Application V4.1 1996

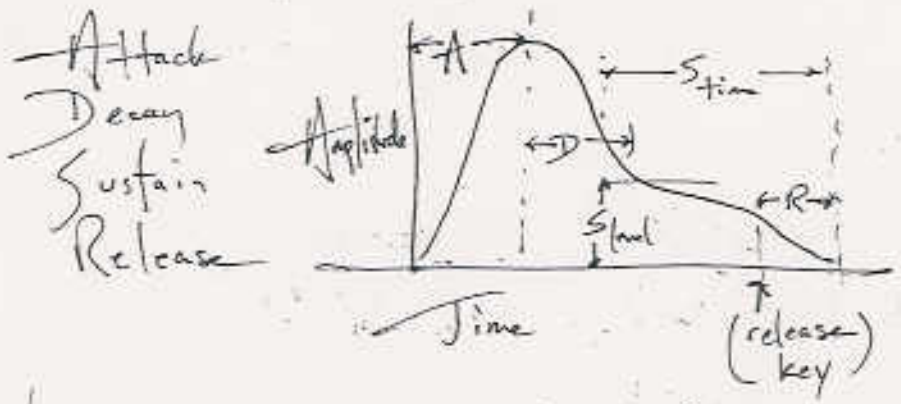
Sunday the 18th, March 19101 at 17:56:24
Filename: C:\DAVIDM~1\SCOPE\ZIONISM
File Commentary: C&R #099184702; Indictment of Zionism. No direct Fourier correlation.



Timebase:	0.5ms/div 20kSa/s
Trigger:	+INTERN\SINGLE
Input Coupling:	AC
Input Voltage:	250mV/div

Delta Time in HERO.26667 KHz

Casio VL-TONE VL-1 — ADSR Waveforming



Voices - 0=Piano; 1=Fantasy; 2=Violin; 3=flute; 4=Guitar 1; 5=Guitar 2
6=English horn; 7=Electro-sound 1; 8=Electro-sound 2; 9=Electro-sound 3.

8 Digits form various voices:

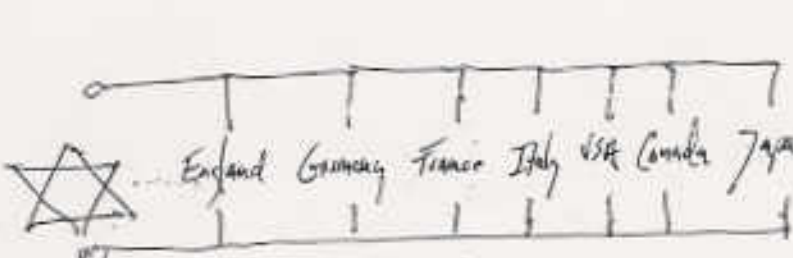
Voice A D S$_L$ S$_t$ R; Vibrato/Tremolo

6 1 0 7 9 1 3 0

Examples - Violin 23099130
Piano 00453200
Synth 70099924
Flute 33099130
English horn 61079130

For instance:

$E_m \sin(2\pi t/T)$ Sine wave.

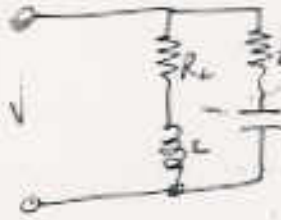


Parallel circuit

Zionism/1948

Simply reduce the resonance

$$f = \frac{1}{2\pi\sqrt{LC}}$$



$$\frac{\frac{1}{2\pi fC}}{R_C^2 + \frac{1}{4\pi^2 f^2 C^2}} - \frac{2\pi fL}{R_L^2 + 4\pi^2 f^2 L^2} = 0$$

So the entire circuit is simplified.

$$V_o(t) = |V| \sin(\omega t + \phi)$$

$$V_o(t) = RI(t) + L\frac{dI(t)}{dt} + \frac{1}{C}\int dI(t)\,dt.$$

the testing has always been a superficial wagering scheme. Eventually the warfare, national debt was to be cancelled.
Cancellation = 180° out of phase.
the beginning of the new state is marked by the end of absurdities.

El Paso County 12/6/99 - CERT# 099184702 began cancellation -
10/27/00 firm is general # 200129251

Market crash coincided with default on 3/14/01.
U.S.D.C. # 01-D-0240

ProbeScope Windows Application V4.1 1996
Sunday the 18th, March 1910I at 18:05:23
Filename: C:\DAVIDM~1\SCOPE\HS4 C.PBS
File Commentary: "MI" at 2x8(11x3)=528 Hz. Cell/genetic healing. "ra gestorum"

Timebase: 0.5ms/div 20kSa/s
Trigger: +INTERN\SINGLE
Input Coupling: AC
Input Voltage: 250mV/div
Delta Time in HERO.52632 KHz

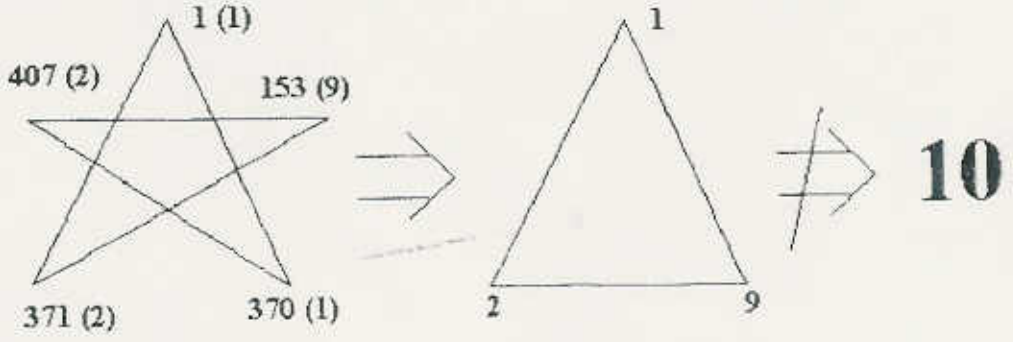
J. Patrick Kelly El Paso Cty,CO 202114298
07/15/2002 09:29
Doc $0.00 Page
Rec $5.00 1 of 1

Fermat's Last Theorem here Proven

Many mathematicians are convinced that Pierre de Fermat (1601-1665) who wrote some of his work in the margins of existing math texts, falsely boasted, "If n is a number greater than two, there are no whole numbers, a, b, c such that $a^n + b^n = c^n$. I have found a truly wonderful proof which this margin is too small to contain." Where Pierre was mistaken is the proof is almost small enough for the margin - which would have avoided many hours of heated debate among mathematicians.

First one should contemplate the only intelligible reason Fermat would have been considering proving a negative, "...there are no whole numbers..." is in the context of Pythagorean Theorem; with a right triangle the sides squared equal the hypotenuse squared. Without this context you might as well prove a zebra is not a giraffe.

Now consider the cube sum number locks. With extremely rare exceptions, any number imaginable will reduce to one of five numbers after repeatedly summing the cubes of the digits. These cube sum number locks are 1, 153, 370, 371 and 407. See for yourself. The key to proving Fermat's (negative question) theorem is to examine the essence of these five numbers. 1 is 1; 153 is 9 (1 + 5 + 3 = 9); 370 is 1 (3 + 7 = 10; 1 + 0 = 1); 371 (11) is 2 and 407 is 2. Now out of the subset of five cube sum number locks we have a subset of three - 1, 2 and 9. Three does not rationalize into ten which is the standard paradigm of our thinking. We think by the decimal system. Ten fingers, ten toes.



The only acceptable "proof" ever developed for Fermat's Last Theorem was executed by computer. Computers "think" in binary. Therefore the obstacle of decimal system thinking was inherently overcome. However, in my opinion, computers are poor witnesses.

00	{506-Byte Prgm} (161 lines)
01	LBL "Stnd"
02	10
03	STO "LWin"
04	286
05	STO "High"
06	LBL 07
07	BEEP
08	RCL "LWin"
09	RCL "Wt"
10	STOP
11	0
12	STO "Accra"
13	STO "Frac"
14	STO "Low"
15	1000
16	STO "Win"
17	CLST
18	1
19	STO "Elem"
20	STO "Incr"
21	LBL 02
22	1
23	INDEX "Isotope"
24	STO IJ
25	RCL EL
26	GTO 00
27	LBL 01
28	RCL "High"
29	RCL "Elem"
30	X≥Y?
31	GTO 10
32	RCL "Incr"

33	1
34	+
35	STO "Incr"
36	GTO 02
37	LBL 00
38	RCL "Elem"
39	1
40	INDEX "Isotope"
41	STO 17
42	RLLEL
43	x≷y
44	CLX
45	+
46	R↑
47	x=y?
48	GTO 01
49	x>y?
50	x≷y
51	÷
52	FP
53	STO "Frac"
54	RCL "Accru"
55	+
56	STO "Accru"
57	RCL "High"
58	RCL "Incr"
59	x<y?
60	GTO 01
61	RCL "Win"
62	RCL "Accru"
63	x<y?
64	GTO 04

66	CLST
67	STO "Accm"
68	STO "Incr"
69	RCL "Elem"
70	1
71	+
72	STO "Elem"
73	RCL "High"
74	X⇄Y
75	X≤Y?
76	GTO 02
77	LBL 04
78	STO "Win"
79	RCL "Elem"
80	STO "Wt"
81	GTO 05
82	LBL 10
83	RCL "LWin"
84	RCL "Wt"
85	STO "LWin"
86	X=Y?
87	GTO 03
88	1
89	INDEX "Isotope"
90	STO 17
91	RCL EL
92	FP
93	STO "Frac"
94	1
95	ENTER
96	STO "Count"

eighty-four of 104

97	RCL "Frac"
98	1
99	−
100	ABS
101	0.5
102	x>y?
103	GTO 08
104	1
105	ENTER
106	LBL 06
107	INDEX "Isotope"
108	STO IJ
109	RCL EL
110	RCL "Frac"
111	−
112	STO EL
113	RCL "Count"
114	1
115	+
116	STO "Count"
117	RCL "High"
118	x<y?
119	GTO 07
120	x⇄y
121	1
122	GTO 06
123	LBL 08
124	1
125	ENTER
126	LBL 09
127	INDEX "Isotope"
128	STO IJ

eighty-five of 101

129	RLLEL
130	RCL "Frac"
131	−
132	1
133	+
134	STOEL
135	RCL "Count"
136	1
137	+
138	STO "Count"
139	RCL "High"
140	X<Y?
141	GTO 07
142	X⇄Y
143	1
144	GTO 09
145	LBL 03
146	RCL "Wt"
147	1
148	INDEX "Isotope"
149	STO 17
150	RCLEL
151	"Isotope"
152	ARCL ST Z
153	AVIEW
154	LBL 11
155	BEEP
156	PSE
157	PSE
158	PSE
159	PSE
160	GTO 11
161	END.

eighty-six of 10

00	{153-Byte Prgm} (93 line program)
01	LBL "Fibona"
02	CLST
03	CLRG
04	FIX 11
05	"Iterations? "F"
06	32
07	STO 04
08	50
09	STO 05
10	1
11	STO 03
12	ENTER
13	Σ-
14	STO 00
15	LBL 00
16	+
17	STO 01
18	RCL 00
19	RCL 01
20	÷
21	100
22	X
23	STO 02
24	STO + 05
25	RCL 03
26	1
27	+
28	STO 03
29	RCL 04
30	-
31	X≤0? eighty-seven of 100
32	GTO 01
33	RCL 05

34	RCL 04
35	1
36	+
37	÷
38	RCL 01
39	RCL 00
40	x<y?
41	x⇄y
42	÷
43	100
44	x
45	x⇄y
46	PSE
47	PSE
48	PSE
49	STO "Basis"
50	RTN
51	LBL 01
52	RCL 00
53	RCL 01
54	+
55	STO 00
56	RCL 01
57	RCL 00
58	÷
59	100
60	x
61	STO 02
62	STO + 05
63	RCL 03
64	1
65	+
66	STO 03

eighty-eight of 99

67	RCL 04
68	−
69	X≤0?
70	GTO 02
71	RCL 05
72	RCL 04
73	1
74	+
75	÷
76	RCL 01
77	RCL 00
78	X<Y?
79	X⇄Y
80	÷
81	100
82	×
83	X⇄Y
84	PSE
85	PSE
86	PSE
87	STO "Basis"
88	RTN
89	LBL 02
90	RCL 01
91	RCL 00
92	GTO 00
93	END

eighty-nine

```
{583-Byte Prgm}   (212 lines)
LBL "Cabala"
CLA
CLST
0
STO "Heb1"
STO "Heb2"
CF 00
"Enter name or -"
⊦"word"
⊦"LF"
PROMPT
LBL "Word"
⊦"▓"
AVIEW
INDEX "Name"
14
ENTER
1
ENTER
LBL 02
STO 17
+
ATOX
STOEL
I+
R↓
X≠Y?
GTO 03
INDEX "Name"
1
ENTER
STO 17
```

Tips - Load "Cabala" and "Rules" last. Program the alpha registers and "Gematra" first instead of keying in the letters over and over.
- Enter the alphabet routines starting with Tav to Aleph.
Load "Heb1", "Heb2" and "Name" - all variables first (before letter values).

To configure "Name"
1[illegible]
ENTER
13 (one less)
[illegible] MATRIX ↓ DIM "NAME"

ninety [illegible]

33	LBL 01
34	RLLEL
35	32
36	$X \rightleftarrows Y$
37	$X = Y$?
38	XEQ "space"
39	65
40	$X \rightleftarrows Y$
41	$X = Y$?
42	XEQ "Aleph"
43	66
44	$X \rightleftarrows Y$
45	$X = Y$?
46	XEQ "Beth"
47	67
48	$X \rightleftarrows Y$
49	$X = Y$?
50	XEQ "Samech"
51	68
52	$X \rightleftarrows Y$
53	$X = Y$?
54	XEQ "Daleth"
55	69
56	$X \rightleftarrows Y$
57	$X = Y$?
58	XEQ "Teth"
59	70
60	$X \rightleftarrows Y$
61	$X = Y$?
62	XEQ "Pe"
63	71
64	$X \rightleftarrows Y$

65	X=Y?
66	XEQ "Gimel"
67	72
68	X⇄Y
69	X=Y?
70	XEQ "He"
71	73
72	X⇄Y
73	X=Y?
74	XEQ "Ayin"
75	74
76	X⇄Y
77	X=Y?
78	XEQ "Yod"
79	75
80	X⇄Y
81	X=Y?
82	XEQ "Caph
83	76
84	X⇄Y
85	X=Y?
86	XEQ "Lamed"
87	77
88	X⇄Y
89	X=Y?
90	XEQ "Mem"
91	78
92	X⇄Y
93	X=Y?
94	XEQ "Nun"
95	79
96	X⇄Y

Ninety-two of 104

97	X=Y?
98	XEQ "Heth"
99	80
100	X⇄Y
101	X=Y
102	XEQ "Pe"
103	81
104	X⇄Y
105	X=Y?
106	XEQ "Koph"
107	82
108	X⇄Y
109	X=Y?
110	XEQ "Resh"
111	83
112	X⇄Y
113	X=Y?
114	XEQ "Shin"
115	84
116	X⇄Y
117	X=Y?
118	XEQ "Tav"
~~119~~	~~67~~
~~120~~	~~X⇄Y~~
~~121~~	~~X=Y?~~
~~122~~	~~XEQ "Samech"~~
123	86
124	X⇄Y
125	X=Y?
126	XEQ "Vau"
127	88
128	X⇄Y

87
X⇄Y
X=Y?
XEQ "Vau"

129	X=Y?
130	XEQ "Tzadoli"
131	89
132	X⇄Y
133	X=Y?
134	XEQ "Yod"
135	90
136	X⇄Y
137	X=Y?
138	XEQ "Zain"
139	I+
140	31
141	X≠Y?
142	GTO 01
143	X=Y?
144	GTO 04
145	LBL 03
146	1
147	X⇄Y
148	GTO 02
149	LBL 04
150	"First word is"
151	↑ "_"
152	ARCL "Heb1"
153	↑ "$_F$"
154	AVIEW
155	PSE
156	PSE
157	FC? 00
158	GTO 05
159	"Second part is_"
160	ARCL "Heb2"

161	⊢"L F"
162	AVIEW
163	PSE
164	PSE
165	PSE
166	LBL05
167	RCL "Heb1"
168	RCL "Heb2"
169	+
170	"Hebrew total_"
171	ARCL ST X
172	⊢"L F"
173	80
174	+
175	286
176	X>Y?
177	GTO 06
178	−
179	286
180	X>Y?
181	GTO 06
182	−
183	286
184	X>Y?
185	GTO 06
186	−
187	286
188	X>Y?
189	GTO 06
190	−
191	286
192	X>Y?

AVIEW

To run program independent of matrix and table of relations

ninety-five of 104

193	GTO 06
194	−
195	R↓
196	LBL 06
197	x⇄y
198	x=0?
199	GTO 07
200	1
201	INDEX "Isotope"
202	STO IJ
203	RCL EL
204	ARCL ST X
205	AVIEW
206	PSE
207	PSE
208	LBL 07
209	0
210	STO "Hel"
211	RTN
212	END

00 {33-byte Prgm} (6 lines)
01 LBL "Aleph"
02 VIEW "Aleph"
03 RCL "Aleph"
04 CLA
PSE
("A" stored in alpha register "Aleph")
05 GTO "Gamatra"
06 END
numerical/consonant features
(index relative weight)
+"Oxen"
AVIEW
PSE

00 {37-Byte Prgm} (8 lines)
01 LBL "Space"
02 "Second word "F"
03 AVIEW
04 PSE
05 CLA
06 0
07 GTO "Gamatra
08 END

Variables - 1st store "Heb1" and "Heb2" and "Name".

Aleph	oxen	1	Lamed	ox goad	30
Beth	house	2	Mem	water	40
Gimel	camel	3	Nun	fish	50
Daleth	door	4	Samek	prop, support	60
He	window	5	Ayin	eye	70
Vav	nail	6	Pe	mouth	80
Zayin	weapon	7	Tsadi	fish hook	90
Heth	fence	8	Qoph	back of head	100
Teth	snake	9	Resh	head	200
Yod	open hand	10	Shin	tooth	300
Kaph	palm of hand	20	Tav	cross	400

```
00  {79-Byte Prgm} (17 lines)
01  LBL "Rules"
02  "Heth is O"F"
03  AVIEW
04  PSE
05  "Teth is E"F"
06  AVIEW
07  PSE
08  "Samech is U"F"
09  AVIEW
10  PSE
11  "Ayin is I"F"
12  AVIEW
13  PSE
14  "Tzaddi is X"F"
15  AVIEW
16  PSE
17  END
18
19  6/15/01 For simplicity - C = samech
20                          K = caph
                            Q = koph
```

00	{50-Byte Prgm} (22 lines)
01	LBL "Gamadra"
02	X⇄Y
03	31
04	X=Y?
05	GTO 05
06	CLX
07	+
08	32
09	X=Y?
10	SF 00
11	CLX
12	+
13	X⇄Y
14	FS? 00
15	GTO 04
16	STO+ "Hel1"
17	GTO 05
18	LBL 04
19	STO+ "Hel2"
20	LBL 05
21	CLX
22	END

00	{413-Byte Prgm} (141 lines)
01	LBL "Matrix"
02	XEQ "Fibona"
03	ALL
04	CLA
05	CLST
06	STO "Hdb1"
07	STO "Hdb2"
08	CF 00
09	1
10	1
11	STO "Count"
12	STO "Number"
13	LBL 01
14	RL
15	1
16	INDEX "Stamina"
17	STO 17
18	RCLEL
19	XTOA
20	XEQ "Word"
21	CLA
22	RCL "Count"
23	1
24	INDEX "stamina"
25	STO 17
26	RCLEL
27	X⇄Y
28	1
29	INDEX "Gate"
30	STO 17
31	RL
32	RL

one hundred of 104

33	STOEL
34	XTOA
35	I+
36	STOEL
37	XTOA
38	XEQ "Word"
39	CLA
40	LBL 04
41	RCL "Number"
42	1
43	INDEX "stamina"
44	STO 17
45	RCLEL
46	X⇄Y
47	1
48	INDEX "Gate"
49	STO 17
50	RV
51	RV
52	STOEL
53	XTOA
54	I+
55	RCLEL
56	XTOA
57	XEQ "Word"
58	CLA
59	1
60	1
61	INDEX "Gate"
62	STO 17
63	RCLEL
64	I+

65	RCLEL
66	1
67	3
68	INDEX "Name"
69	STO17
70	RL
71	RL
72	STOEL
73	I-
74	RL
75	STOEL
76	CLST
77	CLA
78	1
79	1
80	STO "Index"
81	LBL 02
82	X⇌Y
83	1
84	INDEX "Stamina"
85	STO17
86	RCLEL
87	X⇌Y
88	1
89	INDEX "Name"
90	STO17
91	RL
92	RL
93	STOEL
94	XTOA
95	I+
96	RCLEL

97	XTOA
98	I+
99	RLEL
100	XTOA
101	XEQ "Word"
102	CLA
103	1
104	RCL "Index"
105	1
106	+
107	STO "Index"
108	22
109	X≠Y?
110	GTO 02
111	CLA
112	RCL "Number"
113	1
114	+
115	STO "Number"
116	12
117	X⇄Y
118	X≠Y?
119	GTO 04
120	RCL "Number"
121	1
122	STO "Number"
123	RCL "Count"
124	1
125	+
126	STO "Count"
127	22
128	X⇄Y

```
129  X≠Y?
130  GTO 01
131  "Finished"
132  +""
133  ARCL "Basis"
134  AVIEW
135  LBL 10
136  BEEP
137  PSE
138  PSE
139  PSE
140  GTO 10
141  END
```

ATOX

#8

X≠y?

GTO 01 SF 00

LBL 02

STO+ 00

STO "INCR" TEST FS? FC? 00

GTO 01 GTO 03

Golden mean/linear gradient

Fibonacci

61.554% @ 32 iterations

38.5%

Grandpa

Dad — Grandma

50%

50%

Grandpa

Mom — Grandma

Standing wave

gameto-linguistic interface

Contributions increase → exponentially

but effects diminished by time ←.

1973 – Stanley Cohen invents
gene splicing – 3/12 for AB Israel outflows
Human cloning – Ashkenazi are for cloning;
Sephardim against it. Cf p. AB

The protogem is a
Laplace transform; Fourier
series of the extensive
memory – double helix.

⇒ Golden mean.

OPEN
↓
STORE
ITERATIONS
↓
FIRST
ITERATION
AGE – 50
↓
LBLOOD

ADD FIBB
↓
DIVIDE #
X 100
↓ — ADD TO %
TOTAL

INCREMENT
COUNTER

SUBTRACT
~~COUNT TO~~
ITERATIONS
FROM COUNT
COMPARE TO
ZERO

↓
NO
~~DIVIDE~~
~~DISPLAY~~
DIVIDE SUM OF
PERCENTS BY
ITERATIONS
&
DISPLAY

Yes — RECALL FORMULAS

RECALL & ADD
FIBB
↓
DIVIDE #
X 100
↓
ADD TO %
TOTAL
↓
INCREMENT
COUNTER
↓
Subtract ITERATIONS
FROM COUNT COMP TO 0
↓
NO
DIVIDE SUM OF
PERCENTS BY
ITERATIONS & DISPLAY

Yes — RECALL FOLLOWING
SETS