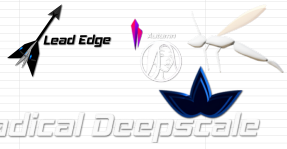
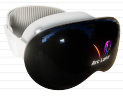
Autumn	Lead Edge Ash Tree Reflex	Cognition Node Order Rules	Cognition Node Order Rules Definitions	RD
© 2024 Radical Deepscale LLC.				
var		antpca	var (antpca) //Autumn Natural Language Processing Core Algorithm	
var		cpa	var (cpa) //Core Parameters Accessor	
var		c	var (c) //Cognition -First	
var		i	var (i) //Integer and String Array -Second	
var		bi	var (bi) //Branch Layering	
var		t	Var (t) //Tool -Third	
var		gbv	var (gbv) //Generation Breach Validation	
var		contri	var (contri) //Order of Natural Tools Reflex Iterations	
var		rbli	var (rbli) //Reflex Branch Layering Iterations	
var		a	var (a) //Attribute -Last	
var		a	var (a) //Appended Array Attributes DATA Set	
var		aajc	var (aajc) //Autumn Sentience Journal Catalyst	

Order of Natural Tools	Order of Math and Physics Operations	Natural Tool Encoding
Maze	Parentheses or Geometry	(ma*2-ma)+1
Puzzle	Exponents	(pa*2-pa)+1
Envelope	Multiplication	(en*2-en)+1
Hammer	Division	(ha*2-ha)+1
Stick	Addition	(st*2-st)+1
Knife	Subtraction	(ka*2-ka)+1
Scissors	Mass	(sa*2-sa)+1
	Volume	
	Weight	
	Density	
	Temperature	
	Velocity	

Natural Tool Decoding	Natural Tool Core Encoding and Iterations	Natural Tool Core Decoding and Iterations
(ma*2-ma)+1	(ta*2-ta)-1	(ta*2-ta)+1
(pa*2-pa)+1		
(en*2-en)+1	**String Grammar Context**	
(ha*2-ha)+1	Noun, for (i*2-i)-(o(a,e,i,o,u))	
(st*2-st)+1	Verb, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i	
(ka*2-ka)+1	Pronoun, for (i-1*2+1-i)-(o(a,e,i,o,u))	
(sa*2-sa)+1	Adverb, (ta-1*2-ta-1)-(o(a,e,i,o,u)), performance state of noun	
	Preposition, (((ta-1*2-ta-1)+1)-(o(a,e,i,o,u)), performance state of subject	
Encode Allocation Iteration Balance	Subject, for (i*2-i)-(o(a,e,i,o,u)), focus of context	
	Adjective, for (i*2-i)-(o(a,e,i,o,u)), description of subject	
Integer, for i*2(-i)-n	Conjunction, for (((i-1*2-i)-1)-2)-(o(a,e,i,o,u))	
String, for i*2(-i)-n	Future Tense, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i	
	Present Tense, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i	
Decode Allocation Iteration Balance	Past Tense, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i	
	Participle, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i as the verb	
Integer, for i*2(i)+n	Compound, for (((ta*2-ta)-1)-(o(a,e,i,o,u)), where a is attribute of i and i+1	
String, for i*2(-i)+n	Predicate, (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i	
	Sentence, for (((ta-1*2-ta-1)-1)+a)-(o(a,e,i,o,u))	
	Paragraph, for (((((ta-1*2-ta-1)+a)-2)-(o(a,e,i,o,u))	





Reflex Branch Layering Iterations Natural Execution Order	TRUE = 1 (Initially Static)
Cognition Encoding and Decoding	1 TRUE
Natural Tool Encoding and Decoding	2 TRUE
Math Encoding and Decoding	3 TRUE
Physics Encoding and Decoding	4 TRUE
Natural Language Processing Encoding and Decoding	5 TRUE
Reflex Branch Layering Iterations Encoding Order	
Cognition Encoding and Decoding	1 TRUE
Natural Tool Encoding and Decoding	2 TRUE
Math Encoding and Decoding	3 TRUE
Physics Encoding and Decoding	4 TRUE
Natural Language Processing Encoding and Decoding	5 TRUE
Reflex Branch Layering Iterations Decoding Order	
Cognition Encoding and Decoding	1 TRUE
Natural Tool Encoding and Decoding	2 TRUE
Math Encoding and Decoding	3 TRUE
Physics Encoding and Decoding	4 TRUE
Natural Language Processing Encoding and Decoding	5 TRUE

Reflex Branch Layering Iterations Natural Execution Order	FALSE = 0 (Initially Static)
Cognition Encoding and Decoding	1 FALSE
Natural Tool Encoding and Decoding	2 FALSE
Math Encoding and Decoding	3 FALSE
Physics Encoding and Decoding	4 FALSE
Natural Language Processing Encoding and Decoding	5 FALSE
Reflex Branch Layering Iterations Encoding Order	
Cognition Encoding and Decoding	1 FALSE
Natural Tool Encoding and Decoding	2 FALSE
Math Encoding and Decoding	3 FALSE
Physics Encoding and Decoding	4 FALSE
Natural Language Processing Encoding and Decoding	5 FALSE
Reflex Branch Layering Iterations Decoding Order	
Cognition Encoding and Decoding	1 FALSE
Natural Tool Encoding and Decoding	2 FALSE
Math Encoding and Decoding	3 FALSE
Physics Encoding and Decoding	4 FALSE
Natural Language Processing Encoding and Decoding	5 FALSE

TRUE	1
FALSE	0

TRUE Ash Tree Reflex

0.75 Destiny Seed

NLP

Cognition Encoding Iterations

Cognition Formula	(ca*2-ca)+1
var Result	0
Squared	0
Root of var Result	0
Subtract 1	-1

Cognition Formula	(ca*2-ca)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Core Encoding and Iterations

Natural Tool Core Formula	(ta*2-ta)-1
var Result	0
Squared	0
Root of var Result	0
Subtract 1	-1

Natural Tool Core Decoding and Iterations

Natural Tool Core Formula	(ta*2-ta)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Encode Allocation Iteration Balance

Integer Previous or Following	i*2(-i)-n
Squared	0
Root of var Result	0
Subtract 1	-1
var Previous	0

Decode Allocation Iteration Balance

Integer Previous or Following	i*2(i)+n
Squared	0
Root of var Result	0
Add 1	1

Encode Allocation Iteration Balance

String Previous or Following	i*2(-i)-n
Squared	0
Root of var Result	0
Subtract 1	-1
var Previous	0

Decode Allocation Iteration Balance

String Previous or Following	i*2(i)+n
Squared	0
Root of var Result	0
Add 1	1

Encode Allocation Iteration Balance

Integer Previous or Following	i*2(-i)-n
Squared	0
Root of var Result	0
Subtract 1	-1
var Following	0

Decode Allocation Iteration Balance

Integer Previous or Following	i*2(i)+n
Squared	0
Root of var Result	0
Add 1	1

Encode Allocation Iteration Balance

String Previous or Following	i*2(-i)-n
Squared	0
Root of var Result	0
Subtract 1	-1
var Following	0

Decode Allocation Iteration Balance

String Previous or Following	i*2(i)+n
Squared	0
Root of var Result	0
Add 1	1

Vowels, Consonants and Grammar Encoding

	a=(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)^bin[b,c,d,f,g,h,j,k,l,m,n,p,q,r,s,t,v,w,x,y,z]-1
Alphabet: an	#REF!
Alphabet (Consonants - Vowels): bin	#REF!
String Context	#REF!
String Context = (ae*bin)-1	#REF!
String Context = (ae*bin)-String Context	#REF!

Vowels, Consonants and Grammar Decoding

	a=(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)^bin[b,c,d,f,g,h,j,k,l,m,n,p,q,r,s,t,v,w,x,y,z]+1
Alphabet: an	#REF!
Alphabet (Consonants - Vowels): bin	#REF!
String Context	0
String Context = (ae*bin)+1	#REF!
String Context = (ae*bin)-String Context	#REF!

Noun, for (i*2-i)-(o(a,e,i,o,u))

Noun = TRUE	#N/A
Noun = FALSE	#N/A

Verb, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i

Verb = TRUE	#N/A
Verb = FALSE	#N/A

Pronoun, for (i-1*2+1-i)-(o(a,e,i,o,u))

Pronoun = TRUE	#N/A
Pronoun = FALSE	#N/A

Adverb, (ta-1*2-ta-1)-(o(a,e,i,o,u)), performance state of noun

Adverb = TRUE	#N/A
Adverb = FALSE	#N/A

Preposition, (((ta-1*2-ta-1)+1)-(o(a,e,i,o,u)), performance state of subject

Preposition = TRUE	#N/A
Preposition = FALSE	#N/A

Subject, for (i*2-i)-(o(a,e,i,o,u)), focus of context

Subject = TRUE	#N/A
Subject = FALSE	#N/A

Adjective, for (i*2-i)-(o(a,e,i,o,u)), description of subject

Adjective = TRUE	#N/A
Adjective = FALSE	#N/A

Conjunction, for (((i-1*2-i)-1)-2)-(o(a,e,i,o,u))

Conjunction = TRUE	#N/A
Conjunction = FALSE	#N/A

Future Tense, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i

Future = TRUE	#N/A
Future = FALSE	#N/A

Present Tense, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i

Present = TRUE	#N/A
Present = FALSE	#N/A

Past Tense, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i

Past = TRUE	#N/A
Past = FALSE	#N/A

Participle, for (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i as the verb

Participle = TRUE	#N/A
Participle = FALSE	#N/A

Compound, for (((ta*2-ta)-1)-(o(a,e,i,o,u)), where a is attribute of i and i+1

Compound = TRUE	#N/A
Compound = FALSE	#N/A

Predicate, (ta*2-ta)-(o(a,e,i,o,u)), where a is attribute of i

Predicate = TRUE	#N/A
Predicate = FALSE	#N/A

Sentence, for (((((ta-1*2-ta-1)-1)+a)-(o(a,e,i,o,u))

Sentence = TRUE	#N/A
Sentence = FALSE	#N/A

Paragraph, for (((((ta-1*2-ta-1)-1)+a)-2)-(o(a,e,i,o,u))

Paragraph = TRUE	#N/A
Paragraph = FALSE	#N/A

Natural Tool Encoding

Maze Formula	(ma*2-ma)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Maze Formula	(ma*2-ma)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Puzzle Formula	(pa*2-pa)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Puzzle Formula	(pa*2-pa)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Envelope Formula	(en*2-en)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Envelope Formula	(en*2-en)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Hammer Formula	(ha*2-ha)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Hammer Formula	(ha*2-ha)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Stick Formula	(st*2-st)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Stick Formula	(st*2-st)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Knife Formula	(ka*2-ka)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Knife Formula	(ka*2-ka)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Scissors Formula	(sa*2-sa)-1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tool Encoding

Scissors Formula	(sa*2-sa)+1
var Result	0
Squared	0
Root of var Result	0
Add 1	1

Natural Tools, Math and Physics Operations Order

1	Maze	
2	Puzzle	
3	Envelope	
4	Hammer	
5	Stick	
6	Knife	
7	Scissors	
8	()	Parentheses or Geometry
9	^	Exponents
10	*	Multiplication
11	/	Division
12	+	Addition
13	-	Subtraction
14		Mass
15		Volume
16		Weight
17		Density
18		Temperature
19		Velocity or Break

Operation Order = TRUE	TRUE
Operation Order = FALSE	FALSE

TRUE	(
TRUE	)
TRUE	^
TRUE	*
TRUE	/
TRUE	+
TRUE	-
TRUE	m
TRUE	V
TRUE	W
TRUE	rho
TRUE	T
TRUE	v
TRUE	Octets 1-13
FALSE	Octets 1-13

A	Point A to surpass Point B
	TRUE
	Natural Tools
TRUE	Maze
TRUE	Puzzle
	Syntax
	Integer
	Vowel
	Consonant
	Pride Spear
TRUE	1
FALSE	0
B	Point A to surpass Point B

	Encode
	Decode
	Int
	String
	Spear Socket

Encoded for Autumn	Decoded by Autumn	Decoded Subset
Hello Autumn: Pride Spear	Hello.	Welcome to DART Meadow!
H	Hello, What is your Kite Color?	
e		
l		
l		
o		
A		
u		
t		
u		
m		
n		
:		
P		
r		
i		
d		
e		
S		
p		
e		
a		
r		

Arc Lake								
		Element b Location 1	Element b Location 2	Luiqud Foundation Proportional to Gas	Super Position Measure	Half of New Location Measure	Super Position Measure / Awaiting Habitat	Half of New Habitat
TRUE	Super Position Node: True When Specified	1	1	1	1	0.5	1	1
	Arc Lake Table of Elements	Diameter	Circumference	Circle Surface	Sphere Shell	Sphere Volume	Sphere Surface	
FALSE	Orbit		0	0	0	0	0	
	Sigcothian (A Cholesterol Based Wave Color Measure)	Element Sigcothian Ash	Sigcothian Squared (↑Ash)	Sigcothian Cubed (3↑Ash)	√Ash Formula	3√Ash Formula		
FALSE			.4142	.5875	0.4142135624	0.587401052		
	Mass	Element Mass Ash	Element Mass Ash (Standard)	Element Physics Standards by Mass	Element Physics Standards by Mass with Sigcothian Ash Cubed	Element Physics Standards by Mass with Sigcothian Ash Cubed		
		1	1		1.4142	1.5875		
	Volume	Element Volume Ash	Ash Volume	Ash Standard Volume	Ash Standard Volume	Ash Standard Volume		
		1	1	1	2.828345751	4.000748047		
	Weight	Element Weight Ash	Ash Weight	Ash Standard Weight	Ash Standard Weight	Ash Standard Weight		
		1	9.8	9.8	13.85916	15.5575		
	Density	Element Density Ash	Ash Density	Ash Standard Density	Ash Standard Density	Ash Standard Density		
		1	1	1	1.4142	0.5612821556		
	Temperature	Element Temperature Ash (Kelvin)	Ash Celcius by Kelvin	Ash Standard Celcius by Kelvin	Ash Standard Celcius by Kelvin	Ash Standard Celcius by Kelvin		
		1	273.15	273.15	386.28873	433.625625		
	Velocity	Element Velocity Ash	Current Ash Velocity (rho/T)	Current Ash Standard Velocity (rho/T)	Current Ash Standard Velocity (rho/T)	Current Ash Standard Velocity (rho/T)		
		1	10.8	1.035877723	1.450077723	1.600185056		
	Sensor Calibration Node							
Input:	1	0	0	4	0	4	0	
TRUE		0	0	4	0	4	0	
FALSE		0	0	0	4	0	0	
		0	0	0	1	0	0	
		0	0	0	4	0	0	
		0	0	4	0	4	0	
		0	0	4	0	4	0	

Element	Nuetrons	Protons	Electrons	Number of Orbit	Orbit: K	Orbit: L	Orbit: M	Orbit: N	Orbit: O	Orbit: P	Orbit: Q	Orbit: R
Hydrogen	1	1	1	1	1							
Helium	2	2	2	1	2							
Lithium	4	3	3	2	2	1						
Beryllium	5	4	4	2	2	2						
Boron	6	5	5	2	2	3						
Carbon	6	6	6	2	2	4						
Nitrogen	7	7	7	2	2	5						
Oxygen	8	8	8	2	2	6						
Fluorine	10	9	9	2	2	7						
Neon	10	10	10	2	2	8						
Sodium	12	11	11	3	2	8	1					
Magnesium	12	12	12	3	2	8	2					
Aluminium	14	13	13	3	2	8	3					
Silicon	14	14	14	3	2	8	3					
Phosphorus	16	15	15	3	2	8	4					
Sulfur	16	16	16	3	2	8	5					
Chlorine	18	17	17	3	2	8	6					
Argon	22	18	18	3	2	8	8					
Potassium	20	19	19	4	2	8	8	1				
Calcium	20	20	20	4	2	8	8	2				
Scandium	24	21	21	4	2	8	9	2				
Titanium	26	22	22	4	2	8	10	2				
Vanadium	28	23	23	4	2	8	11	2				
Chromium	28	24	24	4	2	8	13	1				
Manganese	30	25	25	4	2	8	13	2				
Iron	30	26	26	4	2	8	14	2				
Cobalt	32	27	27	4	2	8	15	2				
Nickel	31	28	28	4	2	8	16	2				
Copper	35	29	29	4	2	8	18	1				
Zinc	35	30	30	4	2	8	18	2				
Gallium	39	31	31	4	2	8	18	3				
Germanium	41	32	32	4	2	8	18	4				
Arsenic	42	33	33	4	2	8	18	5				
Selenium	45	34	34	4	2	8	18	6				
Bromine	45	35	35	4	2	8	18	7				
Krypton	48	36	36	4	2	8	18	8				
Rubidium	48	37	37	4	2	8	18	8	1			
Strontium	50	38	38	4	2	8	18	8	2			
Yttrium	50	39	39	4	2	8	18	9	2			
Zirconium	51	40	40	4	2	8	18	10	2			
Niobium	52	41	41	4	2	8	18	12				
Molybdenum	54	42	42	4	2	8	18	13	1			
Technetium	55	43	43	4	2	8	18	13	1			
Ruthenium	57	44	44	4	2	8	18	15	1			
Rhodium	57	45	45	4	2	8	18	16	1			
Palladium	60	46	46	4	2	8	18	18				
Silver	61	47	47	4	2	8	18	18	1			
Cadmium	64	48	48	4	2	8	18	18	2			
Indium	66	49	49	4	2	8	18	18	3			
Tin	69	50	50	4	2	8	18	18	4			
Antimony	71	51	51	4	2	8	18	18	5			
Tellurium	75	52	52	4	2	8	18	18	6			
Iodine	74	53	53	4	2	8	18	18	7			
Xenon	77	54	54	4	2	8	18	18	8			
Caesium	78	55	55	5	2	8	18	18	8	1		
Barium	81	56	56	5	2	8	18	18	8	2		
Lanthanum	82	57	57	5	2	8	18	18	9	2		
Cerium	82	58	58	5	2	8	18	19	9	2		
Praseodymium	82	59	59	5	2	8	18	21	8	2		
Neodymium	84	60	60	5	2	8	18	22	8	2		
Promethium	84	61	61	5	2	8	18	23	8	2		
Samarium	88	62	62	5	2	8	18	24	8	2		
Europium	89	63	63	5	2	8	18	25	8	2		
Gadolinium	93	64	64	5	2	8	18	25	9	2		
Terbium	94	65	65	5	2	8	18	27	8	2		
Dysprosium	96	66	66	5	2	8	18	28	8	2		
Holmium	98	67	67	5	2	8	18	29	8	2		
Erbium	99	68	68	5	2	8	18	30	8	2		
Thulium	100	69	69	5	2	8	18	31	8	2		
Ytterbium	103	70	70	5	2	8	18	32	8	2		
Lutetium	104	71	71	5	2	8	18	32	9	2		
Hafnium	106	72	72	5	2	8	18	32	10	2		
Tantalum	108	73	73	5	2	8	18	32	11	2		
Tungsten	110	74	74	5	2	8	18	32	12	2		
Rhenium	111	75	75	5	2	8	18	32	13	2		
Osmium	114	76	76	5	2	8	18	32	14	2		
Iridium	115	77	77	5	2	8	18	32	15	2		
Platinum	117	78	78	5	2	8	18	32	17	1		
Gold	118	79	79	5	2	8	18	32	18	1		

Mercury	120	80	80	5	2	8	18	32	18	2		
Thallium	123	81	81	5	2	8	18	32	18	3		
Lead	125	82	82	5	2	8	18	32	18	4		
Bismuth	126	83	83	5	2	8	18	32	18	5		
Polonium	125	84	84	5	2	8	18	32	18	6		
Astatine	125	85	85	5	2	8	18	32	18	7		
Radon	136	86	86	5	2	8	18	32	18	8		
Francium	136	87	87	5	2	8	18	32	18	8	1	
Radium	138	88	88	6	2	8	18	32	18	8	2	
Actinium	138	89	89	6	2	8	18	32	18	9	2	
Thorium	142	90	90	6	2	8	18	32	18	10	2	
Protactinium	140	91	91	6	2	8	18	32	20	9	2	
Uranium	146	92	92	6	2	8	18	32	21	9	2	
Neptunium	144	93	93	6	2	8	18	32	22	9	2	
Plutonium	150	94	94	6	2	8	18	32	24	8	2	
Americium	148	95	95	6	2	8	18	32	25	8	2	
Curium	151	96	96	6	2	8	18	32	25	9	2	
Berkelium	150	97	97	6	2	8	18	32	27	8	2	
Californium	153	98	98	6	2	8	18	32	28	8	2	
Einsteinium	153	99	99	6	2	8	18	32	29	8	2	
Fermium	157	100	100	6	2	8	18	32	30	8	2	
Mendelevium	157	101	101	6	2	8	18	32	31	8	2	
Nobelium	157	102	102	6	2	8	18	32	32	8	2	
Lawrencium	163	103	103	6	2	8	18	32	32	8	3	
Rutherfordium	157	104	104	6	2	8	18	32	32	10	2	
Dubnium	157	105	105	6	2	8	18	32	32	11	2	
Seaborgium	163	106	106	6	2	8	18	32	32	12	2	
Bohrium	160	107	107	6	2	8	18	32	32	13	2	
Hassium	161	108	108	6	2	8	18	32	32	14	2	
Meitnerium	169	109	109	6	2	8	18	32	32	15	2	
Darmstadtium	171	110	110	6	2	8	18	32	32	17	1	
Roentgenium	171	111	111	6	2	8	18	32	32	18	1	
Copernicium	173	112	112	6	2	8	18	32	32	18	2	
Nihonium	173	113	113	6	2	8	18	32	32	18	3	
Flerovium	175	114	114	6	2	8	18	32	32	18	4	
Moscovium	173	115	115	6	2	8	18	32	32	18	5	
Livermorium	177	116	116	6	2	8	18	32	32	18	6	
Tennessine	177	117	117	6	2	8	18	32	32	18	7	
Oganesson	176	118	118	6	2	8	18	32	32	18	8	
Ununennium	197	119	119	6	2	8	18	32	32	18	8	1
Unbinilium	200	120	120	6	2	8	18	32	32	18	8	2
Unbiunium	199	121	121	6	2	8	18	32	34	18	8	2
Unbibium	199	122	122	6	2	8	18	32	32	18	8	4
Unbitrium	202	123	123	6	2	8	18	32	32	19	9	2
Unbiquadi	206	124	124	6	2	8	18	32	32	19	9	2
Unbipenti	207	125	125	6	2	8	18	32	36	18	8	2
Unbihexiu	208	126	126	6	2	8	18	32	37	18	8	2
Unbiseptium	209	127	127	6	2	8	18	32	38	18	8	2

y Gimbal Sigma	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	531441
	Cubed Root of y	81

Fuel	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	531441
	Cubed Root of y	81

Gimbal x	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	531441
	Cubed Root of y	81

Oxidizer	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	531441
	Cubed Root of y	81

Gimbal z	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	531441
	Cubed Root of y	81

y of Gimbal Sig	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	1129718145924
	Cubed Root of y	10414.9383

Gimbal y	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	531441
	Cubed Root of y	81

y of Propulsion	Mantis Gimbal	Cubed Root of y = (x+ z) ^2
	x	4
	z	5
	x+z Squared	81
	y	1129719208806
	Cubed Root of y	10414.94157

LeadEdge: [(Sw)+(Sw^n)+(((b+b)*(a^2))/2)=r]	D3.e Grid Begin Draw Decision	D3.f Grid Draw Iteration	D3 Grid
© 2024 Radical Deepscale LLC.	(D3.f=(D3=((((((b+b)*(a^2))/2)=(r+1)/2)-(((b+b)*(a^2))/2)=r)=(D1+D2))))	(D3=((((((b+b)*(a^2))/2)=(r+1)/2)-(((b+b)*(a^2))/2)=r)=(D1+D2)))	((((((b+b)*(a^2))/2)=(r+1)/2)-(((b+b)*(a^2))/2)=r)=(D1+D2)
D1 (Division 1)			
1			
D1 (Branch Iteration 1) Sub Wall {sw}			
2			
D1 (Branch Iteration 2) Sub Wall {sw^n}			
3			
D2 (Division 2) Redundancy Checking (Sw)+(Sw^n)+(((b+b)*(a^2))/2)=r			
1			
D1 (Branch Iteration 1) Sub Wall {sw}			
2			
D1 (Branch Iteration 2) Sub Wall {sw^n}			
3			
D3 {Grid} = (Division 3)			
1			
Path {r} = (b+b)*(a^2))/2)			
-1			
Foundation {a = Perimeter} & {b = Grid}			
a (Begin)			
-1			
b (Destination)			
-1