Snailz

Melissa Patton and Simon Burleson

Tokens

PLUS	+	MINUS	-
TIMES	*	DIVIDE	/
LPAREN	(	RPAREN	)
NAME	a-z or A-Z or 0-9	NUMBER	0-9
AND	&&	OR	||
SORT	>> (BOGO sort)	LBRA	[
EQUAL	=	RBRA	]

Tokens

COM	,	COMPEQU	==
QUOTE	‘ or “		
PEROID	.		
SCOLIN	;		
GR8R	>		
LES	<		
MOD	%		

Data Types

Shell	tuple (0-9,0-9)
Slow	4 byte 0-9
Slime	8 byte 0-9
Spiral	Yes or No
Snail	a single letter
Escargo	a group of letters

```
Slow var1 = 9;
Slow var2 = 7;
Spiral var3 = (var1=var2);
```

Token parsing of 1st line

```
Token slow Name
Token var1 Name
Token = Equal
Token 9 Number
Token ; Scoln
```

1. This 1st line would store the value 9 in the variable var1
2. This 2nd line would store the value 7 in the variable var2
3. This third line would first evaluate the correctness of the statement and then store No in var3

Slow ex [] = [95,4] >>;

Example of Token parsing

```
Token slow Name
Token ex Name
Token [  Rbra
Token ]  Lbra
Token = equal
Token ]  Lbra
Token 9 Number
Token 5 Number
Token , Com
Token 4 Number
Token , Com
Token ]  Lbra
Token > Sort
Token > Sort
Token ; Scoln
```

1. This would put the slows into an array
2. Then do a bogo sort on the array