Regular Expressions

Digit = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ٩ | ٨ |٧ | ٦ |٥ | ٤ |٣ | ٢ |١ |٠
Unsigned = Digit* Digit
Integer = Unsigned (ε | -)
Real = Unsigned (ε | , Unsigned) (ε | -)
Num = Real | Integer
Letter = غ | ع | ظ | ط | ض | ص | ش | س | ز | ر | | ذ | د | خ | ح | ج | ث | ت | ب | ا |
ف | ق | ك | ل | م | ن | ه | و| ى
ID = (Letter | Digit)* Letter

Tokens

==	=!	=<	=>	=	>
<	؛	،	{	}	(
)	+	-	*	\	ارجع
صحيح	حقيقى	خالى	اذا	اخر	بينما
ID	Num	[	]		

Grammar Rules

1. **program → declaration-list**
2. **declaration-list → declaration declaration-list | declaration**
3. **declaration → var-declaration | fun-declaration**
4. **var-declaration → ؛ ID type-specifier | ؛ [Num] ID type-specifier**
5. **type-specifier → صحيح | حقيقى | خالى**
6. **fun-declaration → compound-stmt (params) ID type-specifier**
7. **params → param-list | ε**
8. **param-list → param-list ، param | param**
9. **param → ID type-specifier | [ ] ID type-specifier**
10. **compound-stmt → { stmt-list local-declarations }**

11. local-declarations → var-declaration local-declarations | ε
12. stmt-list → statement stmt-list | ε
13. statement → expression-statement | compound-statement |
selection-statement | iteration-statement |
return statement
14. expression-stmt → ؛ expression | ؛
15. selection-statement → statement (expression) اذا |
statement اخر statement (expression) اذا
16. iteration-stmt → statement (expression) بينما
17. return-stmt → ارجع ؛ |
ارجع expression ؛
18. expression → expression = var | simple-expression
19. var → ID | [expression] ID
20. simple-expression → additive-expression relOp additive-expression |
additive-expression
21. relOp → =< | => | < | > | =! | ==
22. additive-expression → term addOp additive-expression | term
23. addOp → + | -
24. term → factor mulOp term | factor
**25. mulOp → * | **
26. factor → (expression) | var | call | Num
27. cal → (args) ID
28. args → args-list | ε
29. args-list → expression ، args-list | expression