

BNF Grammar for C-Minus

1. *program* → *declaration-list*
2. *declaration-list* → *declaration-list declaration* / *declaration*
3. *declaration* → *var-declaration* / *fun-declaration*
4. *var-declaration* → *type-specifier ID ;* / *type-specifier ID [NUMENT] ;*
5. *type-specifier* → **int** / **float** / **char** / **void**
6. *fun-declaration* → *type-specifier ID (params) compound-stmt*
7. *params* → *param-list* / **void**
8. *param-list* → *param-list , param* / *param*
9. *param* → *type-specifier ID* / *type-specifier ID []*
10. *compound-stmt* → { *local-declarations statement-list* }
11. *local-declarations* → *local-declarations var-declarations* / *empty*
12. *statement-list* → *statement-list statement* / *empty*
13. *statement* → *expression-stmt* / *compound-stmt* / *selection-stmt* / *iteration-stmt* / *return-stmt*
14. *expression-stmt* → *expression ;* / *;*
15. *selection-stmt* → **if** (*expression*) *statement* / **if** (*expression*) *statement* **else** *statement*
16. *iteration-stmt* → **while** (*expression*) *statement*
17. *return-stmt* → **return ;** / **return** *expression ;*
18. *expression* → *var = expression* / *simple-expression*
19. *var* → **ID** / **ID** [*expression*]
20. *simple-expression* → *additive-expression relop additive-expression* / *additive-expression*
21. *relop* → **<=** / **<** / **>** / **>=** / **==** / **!=**
22. *additive-expression* → *additive-expression addop term* / *term*
23. *addop* → **+** / **-**
24. *term* → *term mulop factor* / *factor*
25. *mulop* → ***** / **/** / **%**
26. *factor* → (*expression*) / *var* / *call* / **NUMENT** / **NUMREAL** / **CHAR**
27. *call* → **ID** (*args*)
28. *args* → *arg-list* / *empty*
29. *arg-list* → *arg-list , expression* / *expression*

TOKENS DEL LENGUAJE

Keywords: **else if int float char return void while**

Special symbols: **+ - * / % < <= > >= == != = ; , () [] { } /* */**

ID = letter (letter| digit|_)*

Comienzan con una letra mayúscula o minúscula, seguida de cualquier cantidad de letras minúsculas, mayúsculas, dígitos o guion bajo

NUMENT = digit digit*

NUMREAL contienen punto y parte decimal

CHAR = un carácter simple encerrado en comillas simples

letter = **a | .. | z | A | .. | Z**

digit = **0 | .. | 9**

Ignorar espacios en blanco y comentarios

Comments: **/* ... */** Varias líneas

// Una sola línea

No terminales se representen en cursivas