Grammar

Non terminal symbols : A , Token, KEYWORD, IDENTIFIER, PUNCTUATOR, CONSTANT, OPERATOR, LITERAL, String

A → Token A |ε

Token → KEYWORD|IDENTIFIER| PUNCTUATOR|CONSTANT|OPERATOR| LITERAL

KEYWORD→ int | void | return
IDENTIFIER→ include | stdio | h | greet
PUNCTUATOR→ # | (|) | { | } | ;
CONSTANT→ 0
OPERATOR→ < | >
LITERAL→ String

Example:

```
#include <stdio.h>

void greet() {
   printf("Hello, World!\n");
}

int main() {

   greet();

   return 0;
}
```

A → Token A
Token A → Token Token A
Token Token A → Token Token Token A
Token Token Token A → Token Token Token Token A
.
.
.

Token Token . . . Token Token A → Token Token . . . Token Token ε - (Token 31 times)

Token Token . . . Token Token ε → PUNCTUATOR IDENTIFIER OPERATOR IDENTIFIER PUNCTUATOR IDENTIFIER OPERATOR KEYWORD IDENTIFIER PUNCTUATOR PUNCTUATOR PUNCTUATOR IDENTIFIER PUNCTUATOR LITERAL PUNCTUATOR PUNCTUATOR PUNCTUATOR KEYWORD IDENTIFIER PUNCTUATOR PUNCTUATOR PUNCTUATOR IDENTIFIER PUNCTUATOR PUNCTUATOR PUNCTUATOR KEYWORD CONSTANT PUNCTUATOR PUNCTUATOR

→ # include < stdio.h > void greet ( ) { printf ("Hello, World!\n") ; } int main ( ) { greet ( ) ; return 0 ; }