<DEM> → # <INC> <DEM> / <START>

<INC> → include < id <LIB> >

<LIB> → . id <LIB> / ε

<START> → <DEC> <START> / <STRUCT> <START> / <MAIN> main <METH>

<MAIN> → <TYPE> / void

<DEC> → <TYPE> id <TDEC> / void id <METH> / id :

<TDEC> → <METH> / <TAB> ;

<METH> → (<P> { <CORPS> }

<P> → <PARAMS>) /)

<PARAMS> → <TYPE> id <PARAM>

<PARAM> → , <PARAMS> / ε

<TAB> → [entier] <TAB2> / <INIT>

<INIT> → = <EA> / ε

<TAB2> → [entier] <INIT2> / <INIT1>

<INIT1> → = <T1> / ε

<T1> → { <T11>

<T11> → <TEA> } / }

<TEA> → <EA> <T12>

<T12> → , <TEA> / ε

<INIT2> →= { <T21>

<T21> → <T2> } / }

<T2> → <T1> <T22>

<T22> → , <T2> / ε

<STRUCT> → struct id { <SDEC> } id ;

<SDEC> → <TYPE> id <TAB> ; <SDEC> / ε

<CORPS>→ <DEC> <SUITE> / <INST> <SUITE> / if <IF> <SUITE> / for <FOR> <SUITE> / while <WHILE> <SUITE> / switch <SWITCH> <SUITE> / goto <GOTO> <SUITE>

<SUITE> → <CORPS> / ε

<INST> → return <VAR> ; / <ID> <OP> ;

<OP> → = <AFFECT> / ++ / --

<AFFECT> → <EA> / <T1> / <T2>

<EA> → (<VAR> <E>) / <VAR> <E>

<E> → <OPA> <EA> / ε

<VAR>→ entier / reel / caractere / chaine / <ID>

<ID> → id <STRD>

<STRD → -> id <STRD> / ε

<OPA>→+ / - / * / /

<IF>→ (<COND>) <EXEC> <ELSE>

<ELSE> → else <EIF> / ε

<EIF> → if <IF> / <EXEC>

<EXEC> → { <CORPS> } / <INST>

<FOR>→(<VE> id = <EA> ; <COND>; id <OPS>) { <CORPS> }

<OPS> → = <EA> / ++ / --

<VE> → int / long / ε

<WHILE> → (<COND>) { <CORPS> }

<COND> → (<COND>) <OL> <CD> <COND2> / <EA> <OPC> <EA> <COND2>

<COND2> → <OL> <CD> <COND2> / ε

<CD> → (<COND>) / <COND>

<TYPE>→ int / long /float / double / char / String

<OPC>→< / > / <= / >= / != / ==

<OPL> → && / ||

<SWITCH> → (<ID>) { <CASES> }

<CASES> → case <VAR> <CASE> <CASES> / default <CASE> / ε

<CASE> → : <CORPS> break ;

<GOTO> → id ;