

programme :  
liste-declarations liste-fonctions

liste-declarations :  
liste-declarations declaration  
| **epsilon**

liste-fontions :  
liste-fonctions fonction  
| **epsilon**

declaration :  
**int** liste-declarateurs ;

liste-declarateurs :  
liste-declarateur , declarateur  
| declarateur

declarateur :  
**identificateur**  
| declarateur [ **constante** ]

fonction :  
type **identificateur** ( liste-parms ) { liste-declarations liste-instructions }  
| **extern** type **identificateur** ( liste-parms ) ;

type :  
**void**  
| **int**

liste-parms : l-parms  
| **epsilon**

l-parms : parm  
| l-parms , parm

liste-instructions :  
liste-instructions instruction  
| **epsilon**

instruction :  
iteration  
| selection  
| saut  
| affectation ;  
| bloc  
| appel

iteration :  
**for** ( affectation ; condition ; affectation ) instruction  
| **while** ( condition ) instruction

selection :  
**if** ( condition ) instruction  
| **if** ( condition ) instruction **else** instruction  
| **switch** ( expression ) instruction  
| **case constante** : instruction  
| **default** : instruction

saut :  
**break** ;  
| **return** ;

	<b>return</b> expression ;
affectation	: variable = expression
bloc	: { liste-declarations liste-instructions }
appel	: <b>identificateur</b> ( liste-expressions ) ;
variable	: <b>identificateur</b>
expression	variable [ expression ] : ( expression )   expression binary-op expression   - expression   <b>constante</b>   variable   <b>identificateur</b> ( liste-expressions )
liste-expressions	: liste-expressions , expression   <b>epsilon</b>
condition	: ! ( condition )   condition binary-rel condition   ( condition )   expression binary-comp expression
binary-op	: +   -   *   /   <<   >>   <b>&amp;</b> 
binary-rel	: <b>&amp;&amp;</b> 
binary-comp	: <   >   >=   <=   ==   !=