<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	→ <func-decs> <main> <func-defs></func-defs></main></func-decs>
<func-decs></func-decs>	→ { <func-dec>} // zero or more function declarations</func-dec>
<func-dec></func-dec>	→ int <point-ref> <id>(<param/>);</id></point-ref>
<point-ref></point-ref>	→ * & lambda
<id></id>	→ <letter>{<letter> <digit>}</digit></letter></letter>
<mut></mut>	→ mut lambda
<subscript></subscript>	→ [] lambda
<main></main>	→ int main() <l-brack><stat-list> return 0;<r-brack></r-brack></stat-list></l-brack>
<stat-list></stat-list>	→ { <statement>}</statement>
<statement></statement>	→ <var-dec> <assign> <while> <if-block> <print>????</print></if-block></while></assign></var-dec>
<func-defs></func-defs>	→ { <func-def>}</func-def>
<func-def></func-def>	→ int <point-ref> <id>(<param/>)<l-brack><stat-list></stat-list></l-brack></id></point-ref>
	<return><r-brack></r-brack></return>
<param/>	→ int <point-ref> <mut> <id><subscript></subscript></id></mut></point-ref>
<return></return>	→ return <ret-val>;</ret-val>
<ret-val></ret-val>	→ <number> <point-ref> <id></id></point-ref></number>
<letter></letter>	→ az
<digit></digit>	→ 09
<var-dec></var-dec>	→ int <point-ref> <mut> <id><array> <dec-rhs>;</dec-rhs></array></id></mut></point-ref>
<assign></assign>	→ <point-ref><id><array>= <assign-rhs>;</assign-rhs></array></id></point-ref>
<while></while>	→ while (<condition>) <l-brack><stat-list><r-brack></r-brack></stat-list></l-brack></condition>
<if-block></if-block>	→ if(<condition>)<l-brack><stat-list>} <elif><else><r-brack></r-brack></else></elif></stat-list></l-brack></condition>
<condition></condition>	→ <not> <id> <cond-op> <cond-rhs></cond-rhs></cond-op></id></not>
<elif></elif>	→ {elif(<condition>)<l-brack><stat-list><r-brack>}</r-brack></stat-list></l-brack></condition>
<else></else>	→ else(<condition>)<l-brack><stat-list><r-brack> lambda</r-brack></stat-list></l-brack></condition>
<l-brack></l-brack>	→ {
<r-brack></r-brack>	→ }
<not></not>	→ {!}????
<cond-op></cond-op>	→ < > == != <= >=
<cond-rhs></cond-rhs>	→ <id> <int></int></id>
<int></int>	→ <sign><digit>{<digit>}</digit></digit></sign>
<sign></sign>	→ - lambda
<assign-rhs></assign-rhs>	→ <expression> <point-ref><id><array> <allocate></allocate></array></id></point-ref></expression>
<dec-rhs></dec-rhs>	→ = <assign-rhs> lambda</assign-rhs>
<array></array>	→ [<int>]</int>
<expression></expression>	→ <expression> + <expression> <factor></factor></expression></expression>
<factor></factor>	→ <point-ref><id><array> <int> (<expression>)</expression></int></array></id></point-ref>
	→