Remove all left recursions			
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	->	program <identifier_start> ; var <dec-list> begin <stat-list> end.</stat-list></dec-list></identifier_start>	
<identifier_start></identifier_start>	->	<letter> <identifier body=""></identifier></letter>	
<identifier_body></identifier_body>	->	<letter> <identifier_body></identifier_body></letter>	
<identifier_body></identifier_body>	->	<digit><identifier_body></identifier_body></digit>	
<identifier_body></identifier_body>	->		
<dec-list></dec-list>	->	<dec> : <type> ;</type></dec>	
<dec></dec>	->		
<dec></dec>	->		
<type></type>	->	_	
<stat-list></stat-list>	->		
<stat-list></stat-list>	->	<stat> <stat-list></stat-list></stat>	
<stat></stat>	->	<write></write>	
<stat></stat>	->	<assign></assign>	
<write></write>	->	display ("value" , <identifier_start>);</identifier_start>	
<write></write>	->	display (<identifier_start>);</identifier_start>	
<assign></assign>	->	<identifier_start> = <expr_term_factor_enter> ;</expr_term_factor_enter></identifier_start>	
<expr_term_factor_enter></expr_term_factor_enter>	->	<term_enter> <expr></expr></term_enter>	
<expr></expr>	->	+ <term_enter> <expr></expr></term_enter>	
<expr></expr>	->	- <term_enter> <expr></expr></term_enter>	
<expr></expr>	->	lamda	
<term_enter></term_enter>	->	<factor> <term></term></factor>	
<term></term>	->	* <factor> <term></term></factor>	
<term></term>	->	/ <factor> <term></term></factor>	
<term></term>	->	lamda	
<factor></factor>	->	(<expr_term_factor_enter>)</expr_term_factor_enter>	
<factor></factor>	->	<id>dentifier_start></id>	
<factor></factor>	->	<number></number>	
<number_start></number_start>	->	<sign> <digit> <number_body></number_body></digit></sign>	
<number_body></number_body>	->	<digit> <number_body></number_body></digit>	
<number_body></number_body>	->	lamda	
<sign></sign>	->	+	
<sign></sign>	->	-	
<sign></sign>	->	lamda	
<digit></digit>	->	0	
<digit></digit>	->	1	
<digit></digit>	->	2	
<digit></digit>	->	3	
<digit></digit>	->	4	
<digit></digit>	_	5	
	->	5	
<digit></digit>	-> ->	6	
<digit></digit>	->	6	
<digit> <digit></digit></digit>	-> ->	6 7	
<digit> <digit> <digit> <digit></digit></digit></digit></digit>	-> -> ->	6 7 8	

<letter></letter>	-> r
<letter></letter>	-> s

cyropy > program <identifier_start> ; var <dec-list> begin <stat-list_enter> end. cidentifier_start> > cletter> <identifier_body> cidentifier_body> > cletter> <identifier_body> cidentifier_body> > clater> <identifier_body> cidentifier_body> > cletter> cidec-list> > cletter> cidec-list > cletter> cidec > landa ctype> > stat-list> cstat-list > cstat-list-list_enter> cstat-list > cstat-list_enter> cstat-list cstat-list_enter> <th>Remove all Indeterminates</th><th></th><th></th></identifier_body></identifier_body></identifier_body></stat-list_enter></dec-list></identifier_start>	Remove all Indeterminates			
cidentifier_start> > < detter> <identifier_body> > < detter> <identifier_body> cidentifier_body> > < digit> cidentifier_body> cidentifier_body> > < landa</identifier_body></identifier_body>	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	->	program <identifier_start> ; var <dec-list> begin <stat-list_enter> end.</stat-list_enter></dec-list></identifier_start>	
cidentifier_body> > < detter> < identifier_body> cidentifier_body> > < digit> < identifier_body> cidentifier_body> > < dec_enter> cidec_enter> > < dec_enter> cidec_enter> > < identifier_start> < dec> cidec> > lamda ctype> > integer cstat-list_enter> > < stat> < stat-list_enter> cstat-list_enter> > stato cstat-list_enter> > cstat-list_enter> cstat-list_enter> > cstat-list_enter> cstat-list_enter> > display (cwrite> cwrite_enter> > display (cwrite> cwrite_enter> > display (cwrite> cwrite_enter> > cidentifier_start> ; cexpr_term_factor_enter> > cterm_enter> > cterm_enter> > cterm_enter> > cterm_enter> > cterm_enter><		->		
cidentifier_body> > cligit> <identifier_body> cidentifier_body> > lamda cdec-list> > cdec_enter> > clidentifier_start> cdec> cdec > , cdec_enter> > clidentifier_start> cdec> cdec> > , cdec_enter> cdec> > lamda ctype> > linteger cstat-list_enter> > cstat> cstat-list> cstat-list> > lamda cstat-list> > cstat> cstat-list_enter> cstat-list> > cwrite_enter> cstat-list> > cwrite_enter> cstat-list > cwrite_enter> cstat-list > cwrite_enter> cstat-list > cwrite_enter> cstat-list > cwrite_enter> cvitete > cwrite_enter> cwrite > cwrite_enter> > cidentifier_start>); cwrite > cidentifier_start>); cwrite > cidentifier_start> cassign> > cidentifier_start cassign> > cidentifier_start cexpr > lamda cterm > cidentifier_start <t< td=""><td></td><td></td><td colspan="2"></td></t<></identifier_body>				
cidentifier_body> > lamda cdec-list> > cdec_enter> : <tytype>; cdec enter> > cdec_enter> cdec> > lamda ctype> > integer cstat-list_enter> > cstat> stat-list> cstat-list> > lamda cstat-list> > lamda cstat-list> > lamda cstat-list> > lamda cstat-list> > cstat>cstat-list_enter> cstat-list> > lamda cstat-list> > cstat-list_enter> cstat-list > cstat-list_enter> cstat-list > cstat-list_enter> cstat > cidentifier_start>; cwrite > cidentifier_start>; cwrite > cidentifier_start> cexpr_texpr_ cexpr_cerm_factor_enter> > cfactor> cfactor> cfactor> cfactor cterm <</tytype>				
cdec enter> > cidentifier_start> cdec> cdec> >, cdec_enter> cdec> > integer ctype> > integer cstat-list_enter> > cstat> cstat-list> cstat-list_one > cstat> cstat-list_enter> cstat-list> > cassign> cstat> > cassign> cwrite_enter> > display (cwrite> cwrite> > cidentifier_start>); cwrite> > cidentifier_start>); cwrite> > cidentifier_start>; cassign> > cidentifier_start>; cexpr > cidentifier_start> = cxpr_term_factor_enter>; cexpr > cidentifier_start> = cxpr_term_factor_enter>; cexpr > cidentifier_start> = cxpr cexpr > cidentifier_start> = cxpr> cexpr > cidentifier_start> = cxpr cexpr > cidentifier_start> = cxpr cexpr > cidentifier_start> cterm > cidentifier_start> cterm > cidentifier_start> cterm > cidentifier_start> cfactor> > cidentifier_start>	<identifier_body></identifier_body>	->		
cdec> > amda cdec> > lamda ctype> > integer cstat-list_enter> > cstat>cstat-list> cstat-list> > amda cstat-list> > amda cstat-list> > cstat>cstat-list_enter> cstat> > cstat> cstat> > cstat>cstat-list_enter> cstat> > cssign> cwrite_enter> > display (swrite> cwrite_enter> > display (swrite> cwrite_enter> > display (swrite> cwrite_enter> > cidentifier_start>); cassign> > cidentifier_start>); cassign> > cidentifier_start>); cexpr term_factor_enter> > cterm_enter> cexpr cexpr> > amda cterm_enter> > cfactor> cterm_enter> > cfactor> cfactor> cfactor> > cfactor> cfactor> cfactor> > cfactor> cfactor > cfactor> cfactor > cfactor cterm> > amda cfactor > cidentifier_start> cfactor > cidentifier_start> cnumber_bod	<dec-list></dec-list>	->	<dec enter="">:<type>;</type></dec>	
cdec> > lamda ctype> > integer cstat-list_enter> > cstat> cstat-list> cstat-list> > lamda cstat-list> > cstat> cstat> cstat> > cwrite_enter> cwrite=enter> > display (cwrite> cwrite> > "value", cldentifier_start>); cwrite> > cidentifier_start>); cwrite> > cidentifier_start>); cwrite> > cidentifier_start>); cwrite- > cidentifier_start>; cwrite- > cidentifier_start>; cwrite- > cidentifier_start> cexpr cexpr > cterm_enter> cexpr> cexpr > cterm_enter> cexpr> cexpr > cterm_enter> cexpr> cexpr > cfactor> cterm> cterm > cfactor> cterm> cterm > cfactor> cterm> cterm > cfactor> cterm cfactor > cidentifier_start> cfactor > cidentifier_start> cfactor > cidentifier_start> cfactor > cidentifier_start> c	<dec_enter></dec_enter>	->		
ctype> > integer cstat-list_enter> > stat> <stat-list> cstat-list> > lamda cstat-list> > stat> <stat-list> cstat> > display (<write> cwrite_enter> > display (<write> cwrite> > "value", cidentifier_start>); cwrite> > identifier_start>); cassign> > cidentifier_start>); cassign> > cidentifier_start>); cassign> > cidentifier_start>); cassign> > cidentifier_start>; cassign> > cidentifier_start> = cexpr_term_factor_enter>; cexpr_term_factor_enter> > cterm_enter> < cexpr> > cterm_enter> < cexpr> cexpr> > lamda cterm> > /<factor> < term> cterm > /<factor> < term> cterm> > imda cterm> > / sidentifier_start> cterm> > imda cterm> > imda cterm> > imda cfactor> > cidentifier_start> cfactor> > cidentifier_start> cfactor></factor></factor></write></write></stat-list></stat-list>	<dec></dec>	->	_	
cstat-list enter> > <stat><stat-list> cstat-list> > lamda cstat-list> > <stat><stat-list_enter> cstat> > cwrite_enter> cstat> > display (swrite> cwrite enter> > display (swrite> cwrite enter> > display (swrite) cwrite enter> > display (swrite) cwrite enter> > cidentifier_start>); cwrite enter> > cidentifier_start>); cwrite enter > cidentifier_start>; cassign> > cidentifier_start> = cexpr_term_factor_enter>; cexpr > + cterm_enter> < cexpr> cexpr > + cterm_enter> < cexpr> cexpr> > lamda cterm > * cfactor> cterm> cterm > /cfactor> cterm> cterm > /cfactor> cterm> cterm > /cfactor> cterm cterm > /cfactor> cterm cterm > /cfactor> cterm cterm > /cfactor> classign cfactor > classign cnumber_start> > classign csign > lamda <</stat-list_enter></stat></stat-list></stat>	<dec></dec>	->		
<stat-list> > lamda <stat-list> > stat> stat-list_enter> <stat> > write_enter> <mrite=enter> > display (swrite> <write> "value", videntifier_start>); <write> > videntifier_start>); <assign> > cidentifier_start> = <expr_term_factor_enter>; <expr_term_factor_enter> > cterm_enter> <expr> <expr> > + cterm_enter> <expr> <expr> > - cterm_enter> <expr> <expr> > - lamda <term> > factor> <term> <term> > /cfactor> <term> <term> > lamda <factor> > cfactor> <term> <term> > lamda <factor> > cfactor> <term> <term> > lamda <factor> > cfector> <term_factor_enter>) <factor> > cidentifier_start> <factor> > cidentifier_start> <number_start> > csign> cdigit><number_body> <number_body> > lamda <sign> > lamda <sign> > lamda <tr< td=""><td><type></type></td><td>-></td><td>integer</td></tr<></sign></sign></number_body></number_body></number_start></factor></factor></term_factor_enter></factor></term></term></factor></term></term></factor></term></term></term></term></term></expr></expr></expr></expr></expr></expr></expr_term_factor_enter></expr_term_factor_enter></assign></write></write></mrite=enter></stat></stat-list></stat-list>	<type></type>	->	integer	
<stat-list> > <stat> <stat-list_enter> <stat> > <mirte_enter> <stat> > <assign> <mrite_enter> > display (swrite> <mrite> > "aulue", <identifier_start>); <mrite> > <identifier_start> = cexpr_term_factor_enter>; <assign> > <identifier_start> = cexpr_term_factor_enter>; <expr_term_factor_enter> > <term_enter> <expr> <expr> > <term_enter> <expr> <expr> > <factor> <term> <expr> > <factor> <term> <term> > <factor> <term> <term> > <term> < <imodesimple <="" <imodesimple="" td=""></imodesimple></term></term></term></term></term></term></term></term></term></term></term></term></term></term></term></term></factor></term></term></factor></expr></term></factor></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter></identifier_start></assign></identifier_start></mrite></identifier_start></mrite></mrite_enter></assign></stat></mirte_enter></stat></stat-list_enter></stat></stat-list>	<stat-list_enter></stat-list_enter>	->	<stat> <stat-list></stat-list></stat>	
<stat> > <mrite_enter> <stat> > <assign> <mrite_enter> > display (<mrite> <mrite> > "value" , <dentifier_start>); <mrite> > identifier_start>); <mrite> > identifier_start> = <expr_term_factor_enter>; <expr_term_factor_enter> > <term_enter> <expr> <expr> > + <term_enter> <expr> <expr> > - <term_enter> <expr> <expr> > - <term_enter> <expr> <expr> > - <term> <term> > <factor> <term> <term> > <factor <term=""> <factor> > <identifier_start> <factor> > <identifier_start> <number_start> > <identifier_start> <number_body> > > <igin> <number_body> > <number_body> > <number_body> > <number_body< td=""> > <number_body< td=""></number_body<></number_body<></number_body></number_body></number_body></igin></number_body></identifier_start></number_start></identifier_start></factor></identifier_start></factor></factor></term></term></factor></term></term></expr></term></expr></term></expr></term></expr></term></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter></expr_term_factor_enter></mrite></mrite></dentifier_start></mrite></mrite></mrite_enter></assign></stat></mrite_enter></stat>	<stat-list></stat-list>	->	lamda	
<stat> > <assign> <write enter=""> > display (swrite> <write> > "value", 'dentifier_start>); <write> > <identifier_start>); <assign> > <identifier_start> = <expr_term_factor_enter>; <expr_term_factor_enter> > <term_enter> <expr> <expr> > + <term_enter> <expr> <expr> > - <term_enter> <expr> <expr> > - <term_enter> <expr> <expr> > lamda <term=enter> > <factor> <erm> <term> > /factor> <term> <term> > /factor> <term> <term> > dentifier_start> <factor> > <identifier_start> <factor> > <idigit> <number_body> <number_start> > <igin> <idigit> <number_body> <number_body> > <la> lamda <sign> > + <sign> > <idigit> > > > <t< td=""><td><stat-list></stat-list></td><td>-></td><td><stat> <stat-list_enter></stat-list_enter></stat></td></t<></idigit></sign></sign></la></number_body></number_body></idigit></igin></number_start></number_body></idigit></factor></identifier_start></factor></term></term></term></term></term></erm></factor></term=enter></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter></expr_term_factor_enter></identifier_start></assign></identifier_start></write></write></write></assign></stat>	<stat-list></stat-list>	->	<stat> <stat-list_enter></stat-list_enter></stat>	
cwrite_enter> -> display (cwrite> cwrite> -> "value" , cidentifier_start>); cwrite> -> cidentifier_start>); cassign> -> cidentifier_start> = cexpr_term_factor_enter>; cexpr_term_factor_enter> -> cterm_enter> cexpr> cexpr> -> - cexpr> -> cterm_enter> cexpr> cexpr> -> lamda cterm= -> cfactor> cterm> cterm> -> cfactor> cterm> cterm> -> lamda cfactor> -> (cexpr_term_factor_enter>) cfactor> -> (dentifier_start> cfactor> -> (identifier_start> cfactor> -> (sign> digit> <number_body> cnumber_start> -> csign> cligit> <number_body> cnumber_body> -> lamda csign> -> lamda cdigit> -> 0 cdigit> -> 0 cdigit> -> 1 cdigit> -> 4 cdigit> -> 6 cdigit> -> 6 cdigit> -> 7</number_body></number_body>	<stat></stat>		_	
cwrite_enter> -> display (cwrite> cwrite> -> "value" , cidentifier_start>); cwrite> -> cidentifier_start>); cassign> -> cidentifier_start> = cexpr_term_factor_enter>; cexpr_term_factor_enter> -> cterm_enter> cexpr> cexpr> -> - cexpr> -> cterm_enter> cexpr> cexpr> -> lamda cterm= -> cfactor> cterm> cterm> -> cfactor> cterm> cterm> -> lamda cfactor> -> (cexpr_term_factor_enter>) cfactor> -> (dentifier_start> cfactor> -> (identifier_start> cfactor> -> (sign> digit> <number_body> cnumber_start> -> csign> cligit> <number_body> cnumber_body> -> lamda csign> -> lamda cdigit> -> 0 cdigit> -> 0 cdigit> -> 1 cdigit> -> 4 cdigit> -> 6 cdigit> -> 6 cdigit> -> 7</number_body></number_body>	<stat></stat>		_	
<write> "value", <identifier_start>); <write> > <identifier_start>); <assign> > <identifier_start> = <expr_term_factor_enter>; <expr_term_factor_enter> > <term_enter> <expr> <expr> > + <term_enter> <expr> <expr> > amda <term_enter> > <factor> <term> <term> > /<factor> <term> <term> > /<factor> <term> <term> > /<factor> <term> <factor> > (<expr_term_factor_enter>) <factor> > <identifier_start> <number_start> > <identifier_start> <number_start> > <identifier_start> <number_body> > <identifier_start> <number_body> > <mbox sign=""> > <tisign> > <mbox sign=""></mbox></tisign></mbox></number_body></identifier_start></number_body></identifier_start></number_start></identifier_start></number_start></identifier_start></factor></identifier_start></factor></identifier_start></factor></identifier_start></factor></identifier_start></factor></identifier_start></factor></identifier_start></factor></expr_term_factor_enter></factor></term></factor></term></term></factor></term></term></factor></term></term></factor></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter></expr_term_factor_enter></identifier_start></assign></identifier_start></write></identifier_start></write>	<write_enter></write_enter>			
cwrite> > <identifier_start>); <assign> > <identifier_start> = <expr_term_factor_enter>; <expr_term_factor_enter> > <term_enter> <expr> <expr> > + <term_enter> <expr> <expr> > - <term_enter> <expr> <expr> > lamda <term> > <factor> <term> <term> > /sfactor> <term> <term> > lamda <term> > lamda <factor> <identifier_start> <factor> > <identifier_start> <factor> > <identifier_start> <number_start> > <identifier_start> <number_body> > <identifier_start> <igip> > <igip> > > <igip> > > > <td>_</td><td></td><td></td></igip></igip></igip></identifier_start></number_body></identifier_start></number_body></identifier_start></number_body></identifier_start></number_body></identifier_start></number_body></identifier_start></number_body></identifier_start></number_start></identifier_start></factor></identifier_start></factor></identifier_start></factor></term></term></term></term></term></factor></term></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter></expr_term_factor_enter></identifier_start></assign></identifier_start>	_			
<expr_term_factor_enter> -> <term_enter> <expr> <expr> -> <term_enter> <expr> <expr> -> <term_enter> <expr> <expr> -> <adctor> <etrm> <term> -> <adctor> <etrm> <term> -> <adctor> <etrm> <term> -> <adctor> <adctor< td=""><td><write></write></td><td></td><td></td></adctor<></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></term></etrm></adctor></term></etrm></adctor></term></etrm></adctor></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter>	<write></write>			
<expr_term_factor_enter> -> <term_enter> <expr> <expr> -> <term_enter> <expr> <expr> -> <term_enter> <expr> <expr> -> <adctor> <etrm> <term> -> <adctor> <etrm> <term> -> <adctor> <etrm> <term> -> <adctor> <adctor< td=""><td><assign></assign></td><td>-></td><td><identifier_start> = <expr_term_factor_enter> ;</expr_term_factor_enter></identifier_start></td></adctor<></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></adctor></term></etrm></adctor></term></etrm></adctor></term></etrm></adctor></expr></expr></term_enter></expr></expr></term_enter></expr></expr></term_enter></expr_term_factor_enter>	<assign></assign>	->	<identifier_start> = <expr_term_factor_enter> ;</expr_term_factor_enter></identifier_start>	
<expr> -> + <term_enter> < expr> <expr> -> - <term_enter> < expr> <term_enter> -> <factor> < term> <term> -> * <factor> < term> <term> -> /<factor> < term> <term> -> amda <factor> -> (<expr_term_factor_enter>) <factor> -> <identifier_start> <factor> -> <inumber_start> <mumber_start> -> <igip> < digit> < number_body> <number_body> -> lamda <sign> -> amda <sign> -> lamda <digit> -> 0 <digit< td=""> -> 2 <digit< td=""> -> 3 <digit< td=""> -> 3 <digit< td=""> -> 4 <digit< td=""> -> 5 <digit< td=""> -> 6 <digit< td=""> -> 6 <digit< td=""> -> 7</digit<></digit<></digit<></digit<></digit<></digit<></digit<></digit<></digit></sign></sign></number_body></igip></mumber_start></inumber_start></factor></identifier_start></factor></expr_term_factor_enter></factor></term></factor></term></factor></term></factor></term_enter></term_enter></expr></term_enter></expr>				
<expr> -> lamda <term_enter> -> <factor> <term> <term> -> * <factor> <term> <term> -> /<factor> <term> <term> -> lamda <factor> -> (<expr_term_factor_enter>) <factor> -> <identifier_start> <factor> -> <number_start> <number_start> -> <sign> <digit> <number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></sign></number_start></number_start></factor></identifier_start></factor></expr_term_factor_enter></factor></term></term></factor></term></term></factor></term></term></factor></term_enter></expr>	<expr></expr>			
<term_enter> -> <factor> <term> <term> -> *cfactor> <term> <term> -> cfactor> <term> <term> -> cfactor> <term < td=""> <factor> -> (cexpr_term_factor_enter>) <factor> -> cidentifier_start> <number_start> -> <siign> <diigit> <number_body> <number_body> -> lamda <sign> -> + <sign> -> amda <digit> -> 0 <digit> -> 1 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></diigit></siign></number_start></factor></factor></term <></term></term></term></term></term></term></factor></term_enter>	<expr></expr>	->	- <term_enter> <expr></expr></term_enter>	
<term> -> * <factor> <term> <term> -> /cfactor> <term> <term> -> lamda <factor> -> (cexpr_term_factor_enter>) <factor> -> <identifier_start> <factor> -> <number_start> <number_start> -> <sign> <digit> <number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></sign></number_start></number_start></factor></identifier_start></factor></factor></term></term></term></term></factor></term>	<expr></expr>	->	lamda	
<term> -> /<factor> <term> <term> -> lamda <factor> -> (expr_term_factor_enter>) <factor> -> <identifier_start> <factor> -> <number_start> <number_start> -> <sign> <digit> <number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></sign></number_start></number_start></factor></identifier_start></factor></factor></term></term></factor></term>	<term_enter></term_enter>	->	<factor> <term></term></factor>	
<term> -> lamda <factor> -> (<expr_term_factor_enter>) <factor> -> <identifier_start> <factor> -> <number_start> <number_start> -> <sign> <digit><number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></sign></number_start></number_start></factor></identifier_start></factor></expr_term_factor_enter></factor></term>	<term></term>	->	* <factor> <term></term></factor>	
<factor> -> (expr_term_factor_enter>) <factor> -> <inumber_start> <number_start> -> <sign> <digit> <number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></sign></number_start></inumber_start></factor></factor>	<term></term>	->	/ <factor> <term></term></factor>	
<factor> -> <identifier_start> <factor> -> <number_start> <number_start> -> <sign> <digit> <number_body> <number_body> -> <lamda< td=""> <sign> -> + <sign> -> <sign> -> <sign> -> <sign> -> <digit> -></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></digit></sign></sign></sign></sign></sign></lamda<></number_body></number_body></digit></sign></number_start></number_start></factor></identifier_start></factor>	<term></term>	->	lamda	
<factor> -> <number_start> <number_body> -> <digit><number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></number_body></number_start></factor>	<factor></factor>	->	(<expr_term_factor_enter>)</expr_term_factor_enter>	
<number_start> -> <sign> <digit> <number_body> <number_body> -> lamda <sign> -> + <sign> -> lamda <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></sign></sign></sign></number_body></number_body></digit></sign></number_start>	<factor></factor>	->	<identifier_start></identifier_start>	
<number_body> -> <digit><number_body> <number_body> -> lamda <sign> -> - <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body></number_body></digit></number_body>	<factor></factor>	->	<number_start></number_start>	
<number_body> -> lamda <sign> -> - <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign></sign></number_body>	<number_start></number_start>	->		
<sign> -> + <sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign></sign>	<number_body></number_body>	->	<digit> <number_body></number_body></digit>	
<sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign>	<number_body></number_body>	->	lamda	
<sign> -> lamda <digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit></sign>	<sign></sign>	->	+	
<digit> -> 0 <digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit></digit>	<sign></sign>	->	-	
<digit> -> 1 <digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit></digit>	<sign></sign>	->	lamda	
<digit> -> 2 <digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit></digit>	<digit></digit>	->	0	
<digit> -> 3 <digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit></digit>	<digit></digit>	->	1	
<digit> -> 4 <digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit></digit>	<digit></digit>	->	2	
<digit> -> 5 <digit> -> 6 <digit> -> 7</digit></digit></digit>	<digit></digit>	->	3	
<digit> -> 6 <digit> -> 7</digit></digit>	<digit></digit>	->	4	
<digit> -> 6 <digit> -> 7</digit></digit>	<digit></digit>	->	5	
		->	6	
<digit> -> 8</digit>	<digit></digit>	->	7	
	<digit></digit>	->	8	

<digit></digit>	-> 9	
<letter></letter>	-> p	
<letter></letter>	-> q	
<letter></letter>	-> r	
<letter></letter>	-> s	