1	<program>	➜	<global_declaration><user_defined_function> act gene () {<function_body>}
2	<global_declaration>	➜	_G <type>; <global_declaration>
3	<global_declaration>	➜	λ
4	<type>	➜	<variables>
5	<type>	➜	clust <clust>
6	<type>	➜	perms <perms>
7	<variables>	➜	dose Identifier <doseval> [1]
8	<variables>	➜	quant Identifier <quantval>
9	<variables>	➜	seq Identifier <seqval>
10	<variables>	➜	allele Identifier <alleleval>
11	<seq>	➜	seqliteral
12	<doseval>	➜	= <dose_lit> <dosevaltail>
13	<doseval>	➜	λ
14	<dosevaltail>	➜	, Identifier <doseval>
15	<dosevaltail>	➜	λ
16	<dose_lit>	➜	doseliteral
17	<dose_lit>	➜	neliteral
18	<quantval>	➜	= <quant_lit> <quantval>
19	<quantval>	➜	, Identifier <quantval>
20	<quantval>	➜	λ
21	<quant_lit>	➜	quantliteral
22	<quant_lit>	➜	nequantliteral
23	<seqval>	➜	= "<seq>" <seqval_tail> [2]
24	<seqval_tail>	➜	, Identifier <seqval_tail> [3]
25	<seqval_tail>	➜	λ
26	<alleleval>	➜	= allelelitera<alleleval>
27	<alleleval>	➜	, Identifier <alleleval>
28	<alleleval>	➜	λ
29	<perms>	➜	<perms_variables> [4]
30	<perms_variables>	➜	dose Identifier<dose_perms_val>
31	<perms_variables>	➜	quant Identifier <quant_perms_val>
32	<perms_variables>	➜	seq Identifier<seq_perms_val>
33	<perms_variables>	➜	allele Identifier<allele_perms_val>
34	<dose_perms_val>	➜	= <dose_lit> <dose_perms_val_tail>
35	<dose_perms_val_tail>	➜	, Identifier <dose_perms_val>

36	<quant_perms_val>	➜	= <quant_lit> <quant_perms_val_tail>
37	<quant_perms_val_tail>	➜	, Identifier <quant_perms_val>
38	<seq_perms_val>	➜	= "<seq>" <seq_perms_val_tail> [5]
39	<seq_perms_val_tail>	➜	, Identifier <seq_perms_val>
40	<allele_perms_val>	➜	= alleleLit <allele_perms_val_tail>
41	<allele_perms_val_tail>	➜	, Identifier <allele_perms_val>
42	<clust>	➜	dose Identifier [doseliteral]<optional_dose_2d>
43	<optional_dose_2d>	➜	[doseliteral] <option_in_dose_2d>
44	<optional_dose_2d>	➜	= {<dose_clust>}
45	<optional_dose_2d>	➜	λ
46	<option_in_dose_2d>	➜	= {<twoDdose_clust>}
47	<option_in_dose_2d>	➜	λ
48	<dose_clust>	➜	<dose_lit> <dose_clust_tail>
49	<dose_clust_tail>	➜	, <dose_clust>
50	<dose_clust_tail>	➜	λ
51	<dose_clust>	➜	<twoDdose_clust>
52	<twoDdose_clust>	➜	{<dose_clust>} <twoDdose_clust_tail>
53	<twoDdose_clust_tail>	➜	, <twoDdose_clust>
54	<twoDdose_clust_tail>	➜	λ
55	<clust>	➜	quant Identifier [doseliteral]<optional_quant_2d>
56	<optional_quant_2d>	➜	[doseliteral] <option_in_quant_2d>
57	<optional_quant_2d>	➜	= {<quant_clust>}
58	<optional_quant_2d>	➜	λ
59	<option_in_quant_2d>	➜	= {<twoDquant_clust>}
60	<option_in_quant_2d>	➜	λ
61	<quant_clust>	➜	<quant_lit> <quant_clust_tail>
62	<quant_clust_tail>	➜	, <quant_clust>
63	<quant_clust_tail>	➜	λ
64	<quant_clust>	➜	<twoDquant_clust>
65	<twoDquant_clust>	➜	{<quant_clust>} <twoDquant_clust_tail>
66	<twoDquant_clust_tail>	➜	, <twoDquant_clust>
67	<twoDquant_clust_tail>	➜	λ
68	<clust>	➜	seq Identifier [doseliteral]<optional_seq_2d>
69	<optional_seq_2d>	➜	[doseliteral] <option_in_seq_2d>
70	<optional_seq_2d>	➜	= {<seq_clust>}

71	<optional_seq_2d>	➜	λ
72	<option_in_seq_2d>	➜	= {<twoDseq_clust>}
73	<option_in_seq_2d>	➜	λ
74	<seq_clust>	➜	"<seq>" <seq_clust_tail>
75	<seq_clust_tail>	➜	, <seq_clust>
76	<seq_clust_tail>	➜	λ
77	<seq_clust>	➜	<twoDseq_clust>
78	<twoDseq_clust>	➜	{<seq_clust>} <twoDseq_clust_tail>
79	<twoDseq_clust_tail>	➜	, <twoDseq_clust>
80	<twoDseq_clust_tail>	➜	λ
81	<user_defined_function>	➜	act <function_type> Identifier (<params>){<function_body>}
82	<function_type>	➜	λ
83	<function_type>	➜	void
84	<params>	➜	<param_datatype> Identifier <paramtail>
85	<params>	➜	λ
86	<paramtail>	➜	,<param_datatype> Identifier<paramtail>
87	<paramtail>	➜	λ
88	<param_datatype>	➜	dose
89	<param_datatype>	➜	quant
90	<param_datatype>	➜	seq
91	<param_datatype>	➜	allele
92	<function_body>	➜	<local_declaration> <function_body>
93	<function_body>	➜	<body_statements> <function_body>
94	<function_body>	➜	λ
95	<local_declaration>	➜	_L <type>; <local_declaration>
96	<local_declaration>	➜	λ
97	<body_statements>	➜	<body_statements><body_statements>
98	<body_statements>	➜	λ
99	<body_statements>	➜	<if_statement> <body_statements>
100	<body_statements>	➜	<while_statement> <body_statements>
101	<body_statements>	➜	<for_loop_statement><body_statements>
102	<body_statements>	➜	<do_while_statement> <body_statements>
103	<body_statements>	➜	<express_statement> <body_statements>
104	<body_statements>	➜	<stimuli_statement> <body_statements>
105	<body_statements>	➜	<assignment_statement> <body_statements>

106	<express_statement>	➜	express(<express_value>)
107	<stimuli_statement>	➜	stimuli (<stimuli_value>)
108	<stimuli_value>	➜	seqliteral
109	<destroy_statement>	➜	destroy; <body_statement_tail>
110	<contig_statement>	➜	contig;
111	<prod_statement>	➜	prod <prod_value>;
112	<prod_statement>	➜	λ
113	<prod_value>	➜	Identifier
114	<prod_value>	➜	<literals>
115	<prod_value>	➜	<arithmetic_sequence>
116	<prod_value>	➜	λ
117	<express_statement>	➜	express(<express_value>)
118	<express_value>	➜	<literals> <express_value>
119	<express_value>	➜	Identifier <express_value>
120	<express_value>	➜	<seq_concat> <express_value>
121	<express_value>	➜	<seq_type_cast> <express_value>
122	<express_value>	➜	<arithmetic_operation> <express_value>
123	<express_value>	➜	Identifier [<doseliteral>] <express_value>
124	<express_value>	➜	λ
125	<stimuli_statement>	➜	stimuli (<stimuli_value>)
126	<stimuli_value>	➜	seqliteral
127	<destroy_statement>	➜	destroy;
128	<contig_statement>	➜	contig;
129	<assignment_statement>	➜	Identifier += <assignment_value>
130	<assignment_statement>	➜	Identifier *= <assignment_value>
131	<assignment_statement>	➜	Identifier -= <assignment_value>
132	<assignment_statement>	➜	Identifier /= <assignment_value>
133	<assignment_statement>	➜	Identifier %= <assignment_value>
134	<assignment_statement>	➜	Identifier = <assignment_value>
135	<assignment_value>	➜	<literals>
136	<assignment_value>	➜	for(<initialization> ; <condition>; <update>){<body_statements>} <body_statements>
137	<assignment_value>	➜	<arithmetic_operation>
138	<if_statement>	➜	if (<conditional_block>) {<body_statements>}<if_tail>
139	<if_tail>	➜	<elif_clause>
140	<if_tail>	➜	<else_clause>

141	<if_tail>	➜	λ
142	<conditional_block>	➜	<conditions_base>
143	<conditions_base>	➜	<conditon_value><relational_operator><conditon_value><conditions_logic>
144	<conditions_base>	➜	<conditon_value><relational_operator><conditon_value><conditions_logic>
145	<condition_logic>	➜	λ
146	<condition_logic>	➜	<logical_operators><conditions_base>
147	<condition_value>	➜	<arithmetic_sequence>
148	<condition_value>	➜	Identifiers
149	<condition_value>	➜	<literals>
150	<elif_clause>	➜	<elif_clause>(conditional_block){<body_statements>}<elif_clause><else_clause>
151	<elif_clause>	➜	λ
152	<else_clause>	➜	else{<body_statements>}
153	<else_clause>	➜	λ
154	<relational_operator>	➜	<
155	<relational_operator>	➜	>
156	<relational_operator>	➜	<=
157	<relational_operator>	➜	=>
158	<relational_operator>	➜	==
159	<relational_operator>	➜	!=
160	<seq_concat>	➜	<seq_concat_value> + <seq_concat_value> <seq_concat_tail>
161	<seq_concat_value>	➜	<seq>
162	<seq_concat_value>	➜	<seq_type_cast>
163	<seq_type_cast>	➜	seq(<doseval>)
164	<seq_type_cast>	➜	seq(<quantval>)
165	<seq_type_cast>	➜	seq(Identifier)
166	<seq_type_cast>	➜	λ
167	<seq_concat_tail>	➜	+ <seq_concat_value><seq_concat_tail>
168	<seq_concat_tail>	➜	λ
169	<math_operator>	➜	+
170	<math_operator>	➜	*
171	<math_operator>	➜	-
172	<math_operator>	➜	/
173	<arithmetic_sequence>	➜	<math_lit><math_operator><math_lit>><arithmetic_sequence_tail>
174	<arithmetic_sequence_tail>	➜	λ
175	<arithmetic_sequence_tail>	➜	<arithmetic_sequence>

176	<math_lit>	➜	<dose_lit>
177	<math_lit>	➜	<quant_lit>
178	<do_while_statment>	➜	do{<body_statements><body_statements_tail>}<while_statement>
179	<while_statement>	➜	while(<conditional_block>){<body_statements><body_statements_tail>}
180	<for_statement>	➜	for(<initialization> ; <condition>; <update>){<body_statements>} <body_statements>
181	<initialization>	➜	dose Identifier = <init_value>
182	<init_value>	➜	<dose_lit>
183	<init_value>	➜	Identifier
184	<condition>	➜	Identifier <relational_op> <dose_lit>
185	<update>	➜	<unary_operator> Identifier
186	<update>	➜	Identifier <unary_operator>
187	<unary_operator>	➜	++
188	<unary_operator>	➜	--
189	<literals>	➜	doseliteral
190	<literals>	➜	seqliteral
191	<literals>	➜	quantliteral
192	<literals>	➜	alleleliteral
193	<literals>	➜	neliteral
194	<literals>	➜	nequantliteral

[1] sample
integerslit = positive number without decimal
NeInt.. = negative integerslit

X (strint_ident)
10 and empty or null(intval)

dose X;
dose X = 10;
dose X = ^10;

[2] same kasing string lit sa string ident so irereuse ko yung string ident

[3] same kasing string lit sa string ident so irereuse ko yung string ident

[4] Kung ano laman ng variables with addition lang ng perms sa unahan

[5] same kasing string lit sa string ident so irereuse ko yung string ident