ID	Name	First	Leads	Follows
1.	*program*	**program**		$
2.1.	*identifier_list*	**id**		)
2.2.1.	*identifier_list′*	**,**		)
2.2.2.	*identifier_list′*	ϵ	→	)
3.1.	*declarations*	**var**		**procedure begin**
3.2.	*declarations*	ϵ	→	**procedure begin**
4.1.	*type*	**integer real**		**;)**
4.2.	*type*	**array**		**;)**
5.1.	*standard_type*	**integer**		**;)**
5.2.	*standard_type*	**real**		**;)**
6.1.	*subprogram_declarations*	**procedure**		**begin**
6.2.	*subprogram_declarations*	ϵ	→	**begin**
7.	*subprogram_declaration*	**procedure**		**;**
8.	*subprogram_head*	**procedure**		**var procedure begin**
9.1.	*arguments*	**(**		**;**
9.2.	*arguments*	ϵ	→	**;**
10.1.	*parameter_list*	**id**		**)**
10.2.1.	*parameter_list′*	**;**		**)**
10.2.2.	*parameter_list′*	ϵ	→	**)**
11.	*compound_statement*	**begin**		**; . end else**
12.1.	*optional_statements*	**id call begin while if**		**end**
12.2.	*optional_statements*	ϵ	→	**end**
13.1.	*statement_list*	**id call begin while if**		**end**
13.2.1.	*statement_list′*	**;**		**end**
13.2.2.	*statement_list′*	ϵ	→	**end**
14.1.	*statement*	**id**		**; end else**
14.2.	*statement*	**call**		**; end else**
14.3.	*statement*	**begin**		**; end else**
14.4.	*statement*	**while**		**; end else**
14.5.	*statement*	**if**		**; end else**
15.1.	*else′*	**else**		**; end else**
15.2.	*else′*	ϵ	→	**; end else**
16.	*variable*	**id**		**assignop**
17.1.	*array_access*	**[**		**assignop**
17.2.	*array_access*	ϵ	→	**assignop**
18.	*procedure_statement*	**call**		**; end else**
19.1.	*optional_expressions*	**(**		**; end else**
19.2.	*optional_expressions*	ϵ	→	**; end else**
20.1.	*expression_list*	**id num (not + -**		**)**
20.2.1.	*expression_list′*	**,**		**)**
20.2.2.	*expression_list′*	ϵ	→	**)**
21.	*expression*	**id num (not + -**		**; end else do then]) ,**
22.1.	*related_expression*	**relop**		**; end else do then]) ,**
22.2.	*related_expression*	ϵ	→	**; end else do then]) ,**
23.1.1.	*simple_expression*	**id num (not**		**relop ; end else do then]) ,**
23.1.2.	*simple_expression*	**+ -**		**relop ; end else do then]) ,**
23.2.1.	*simple_expression′*	**addop**		**relop ; end else do then]) ,**
23.2.2.	*simple_expression′*	ϵ	→	**relop ; end else do then]) ,**
24.1.	*term*	**id num (not**		**addop relop ; end else do then]) ,**
24.2.1.	*term′*	**mulop**		**addop relop ; end else do then]) ,**
24.2.2.	*term′*	ϵ	→	**addop relop ; end else do then]) ,**
25.1.1.	*factor*	**id**		**mulop addop relop ; end else do then]) ,**
25.1.2.	*factor*	**num**		**mulop addop relop ; end else do then]) ,**
25.1.3.	*factor*	**(**		**mulop addop relop ; end else do then]) ,**
25.1.4.	*factor*	**not**		**mulop addop relop ; end else do then]) ,**
25.2.1.	*factor′*	**[**		**mulop addop relop ; end else do then]) ,**
25.2.1.	*factor′*	ϵ	→	**mulop addop relop ; end else do then]) ,**
26.1.	*sign*	**+**		**id num not (**
26.2.	*sign*	**-**		**id num not (**