

[1]	<PROGRAM>	→	<IDENT><ENV><DATA><PROC>
[2]	<IDENT>	→	identification division <END_INST> program-id. ID<END_INST> author. WORDS<END_INST> date-written. WORDS <END_INST>
[3]	<WORDS>	→	ID <WORDS_LR>
[4]	<WORDS_LR>	→	ID <WORDS_LR>
[5]		→	ε
[6]	<END_INST>	→	.\n
[7]	<ENV>	→	environment division<END_INST> configuration section<END_INST> source-computer. WORDS<END_INST> object-computer. WORDS<END_INST>
[8]	<DATA>	→	data division<END_INST> working-storage section<END_INST> <VAR_LIST>
[9]	<VAR_LIST>	→	<VAR_DECL> <VAR_LIST>
[10]		→	ε
[11]	<VAR_DECL>	→	<LEVEL> ID pic IMAGE <VAR_DECL_TAIL>
[12]	<VAR_DECL_TAIL>	→	value INTEGER<END_INST>
[13]		→	<END_INST>
[14]	<LEVEL>	→	INTEGER
[15]	<PROC>	→	procedure division<END_INST> ID section<END_INST> <LABELS> end program ID.
[16]	<LABELS>	→	<LABEL><END_INST> <INSTRUCTION_LIST> <LABELS_LR>
[17]	<LABELS_LR>	→	<LABEL><END_INST> <INSTRUCTION_LIST> <LABELS_LR>
[18]		→	ε
[19]	<LABEL>	→	ID
[20]	<INSTRUCTION_LIST>	→	<INSTRUCTION> <INSTRUCTIONLIST>
[21]		→	ε
[22]	<INSTRUCTION>	→	<ASSIGNATION>
[23]		→	<IF>
[24]		→	<CALL>
[25]		→	<READ>
[26]		→	<WRITE>
[27]		→	stop run<END_INST>
[28]	<ASSIGNATION>	→	move <EXPRESSION> to ID<END_INST>
[29]		→	compute ID = <EXPRESSION><END_INST>
[30]		→	add <EXPRESSION> to ID<END_INST>
[31]		→	subtract <EXPRESSION> from ID<END_INST>
[32]		→	multiply <ASSING_END><END_INST>
[33]		→	divide <ASSING_END><END_INST>
[34]	<ASSIGN_END>	→	<EXPRESSION>,<EXPRESSION> giving ID
[35]	<EXPRESSION>	→	<EXP_AND> <EXPRESSION_LR>
[36]	<EXPRESSION_LR>	→	or <EXP_AND> <EXPRESSION_LR>
[37]		→	ε
[38]	<EXP_AND>	→	<EXP_EQUAL> <EXP_AND_LR>
[39]	<EXP_AND_LR>	→	and <EXP_EQUAL> <EXP_AND_LR>

[40]		→ ε
[41]	<EXP_EQUAL>	→ <EXP_ADD> <EXP_EQUAL_LR>
[42]	<EXP_EQUAL_LR>	→ = <EXP_ADD> <EXP_EQUAL_LR>
[43]		→ < <EXP_ADD> <EXP_EQUAL_LR>
[44]		→ > <EXP_ADD> <EXP_EQUAL_LR>
[45]		→ <= <EXP_ADD> <EXP_EQUAL_LR>
[46]		→ >= <EXP_ADD> <EXP_EQUAL_LR>
[47]		→ ε
[48]	<EXP_ADD>	→ <EXP_MULT> <EXP_ADD_LR>
[49]	<EXP_ADD_LR>	→ + <EXP_MULT> <EXP_ADD_LR>
[50]		→ - <EXP_MULT> <EXP_ADD_LR>
[51]		→ ε
[52]	<EXP_MULT>	→ <EXP_NOT> <EXP_MULT_LR>
[53]	<EXP_MULT_LR>	→ * <EXP_NOT> <EXP_MULT_LR>
[54]		→ / <EXP_NOT> <EXP_MULT_LR>
[55]		→ ε
[56]	<EXP_NOT>	→ -<EXP_NOT>
[57]		→ not <EXP_NOT>
[58]		→ <EXP_PARENTHESIS>
[59]	<EXP_PARENTHESIS>	→ (<EXPRESSION>)
[60]		→ <EXP_TERM>
[61]	<EXP_TERM>	→ ID
[62]		→ INTEGER
[63]		→ true
[64]		→ false
[65]	<IF>	→ if <EXPRESSION> then <INSTRUCTION_LIST> <IF_END>
[66]	<IF_END>	→ else <INSTRUCTION_LIST> end-if
[67]		→ end-if
[68]	<CALL>	→ perform ID <CALL_TAIL>
[69]	<CALL_TAIL>	→ until <EXPRESSION><END_INST>
[70]		→ <END_INST>
[71]	<READ>	→ accept ID<END_INST>
[72]	<WRITE>	→ display <WRITE_TAIL>
[73]	<WRITE_TAIL>	→ <EXPRESSION><END_INST>
[74]		→ STRING<END_INST>

Variable	First ¹	Follow ¹
<PROGRAM>	identification	
<IDENT>	identification	
<WORDS>	ID	
<WORDS_LR>	ID, ε	
<END_INST>	.	
<ENV>	environment	
<DATA>	data	
<VAR_LIST>	INTEGER, ε	
<VAR_DECL>	INTEGER	
<VAR_DECL_TAIL>	value, .	

<LEVEL>	INTEGER
<PROC>	procedure
<LABELS>	ID
<LABELS_LR>	ID, ε
<LABEL>	ID
<INSTRUCTION_LIST>	move, compute, add, subtract, multiply, divide, if, perform, accept, display, stop, ε
<INSTRUCTION>	move, compute, add, subtract, multiply, divide, if, perform, accept, display, stop
<ASSIGNATION>	move, compute, add, subtract, multiply, divide
<ASSIGN_END>	-, not, (, ID, INTEGER, true, false
<EXPRESSION>	-, not, (, ID, INTEGER, true, false
<EXPRESSION_LR>	or, ε
<EXP_AND>	-, not, (, ID, INTEGER, true, false
<EXP_AND_LR>	and, ε
<EXP_EQUAL>	-, not, (, ID, INTEGER, true, false
<EXP_EQUAL_LR>	=, <, >, <=, >=, ε
<EXP_ADD>	-, not, (, ID, INTEGER, true, false
<EXP_ADD_LR>	+, -, ε
<EXP_MULT>	-, not, (, ID, INTEGER, true, false
<EXP_MULT_LR>	*, /, ε
<EXP_NOT>	-, not, (, ID, INTEGER, true, false
<EXP_PARENTHESIS>	(, ID, INTEGER, true, false
<EXP_TERM>	ID, INTEGER, true, false
<IF>	if
<IF_END>	else, end-if
<CALL>	perform
<CALL_TAIL>	until, .
<READ>	accept
<WRITE>	display
<WRITE_TAIL>	STRING, -, not, (, ID, INTEGER, true, false