[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> <INSTRUCTION_LIST>
[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 <ASSIGN_END><END_INST>
[33] → divide <ASSIGN_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	first(ENV)
<WORDS>	ID	first(END_INST)
<WORDS_LR>	ID, ε	follow(WORDS)

<END_INST>	.	program-id, date-written, follow(IDENT), configuration, source-computer, object-computer, follow(ENV), working-storage, follow(VAR_DELC_TAIL>), ID, first(LABELS), first(INSTRUCTION_LIST), follow(INSTRUCTION), follow(ASSIGNATION), follow(CALL_TAIL), follow(WRITE_TAIL, READ)
<ENV>	environment	first(DATA)
<DATA>	data	first(PROC)
<VAR_LIST>	INTEGER, ε	follow(DATA)
<VAR_DECL>	INTEGER	first(VAR_LIST)
<VAR_DECL_TAIL>	value, .	follow(VAR_DECL)
<LEVEL>	INTEGER	ID
<PROC>	procedure	follow(PROGRAM)
<LABELS>	ID	end
<LABELS_LR>	ID, ε	follow(LABELS)
<LABEL>	ID	first(END_INST)
<INSTRUCTION_LIST>	move, compute, add, substract, multiply, divide, if, perform, accept, display, stop, ε	first(LABELS_LR)
<INSTRUCTION>	move, compute, add, substract, multiply, divide, if, perform, accept, display, stop	first(INSTRUCTION_LIST)
<ASSIGNATION>	move, compute, add, substract, multiply, divide	follow(INSTRUCTION)
<ASSIGN_END>	-, not, (, ID, INTEGER, true, false	first(END_INST)
<EXPRESSION>	-, not, (, ID, INTEGER, true, false	to, first(END_INST), from, , , giving,) then
<EXPRESSION_LR>	or, ε	follow(EXPRESSION)
<EXP_AND>	-, not, (, ID, INTEGER, true, false	first(EXPRESSION_LR)
<EXP_AND_LR>	and, ε	follow(EXP_AND)
<EXP_EQUAL>	-, not, (, ID, INTEGER, true, false	first(EXP_AND_LR)
<EXP_EQUAL_LR>	=, <, >, <=, >=, ε	follow(EXP_EQUAL)
<EXP_ADD>	-, not, (, ID, INTEGER, true, false	first(EXP_EQUAL_LR)
<EXP_ADD_LR>	+, -, ε	follow(EXP_ADD)
<EXP_MULT>	-, not, (, ID, INTEGER, true, false	first(EXP_ADD_LR)
<EXP_MULT_LR>	*, /, ε	follow(EXP_MULT)
<EXP_NOT>	-, not, (, ID, INTEGER, true, false	first(EXP_MULT_LR)
<EXP_PARENTHESIS>	(, ID, INTEGER, true, false	follow(EXP_NOT)
<EXP_TERM>	ID, INTEGER, true, false	follow(EXP_PARENTHESIS)
<IF>	if	follow(INSTRUCTION)
<IF_END>	else, end-if	follow(IF)
<CALL>	perform	follow(INSTRUCTION)
<CALL_TAIL>	until, .	follow(CALL)

<READ>	accept	follow(INSTRUCTION)
<WRITE>	display	follow(INSTRUCTION)
<WRITE_TAIL>	STRING, -, not, (, ID, INTEGER, true, false	follow(WRITE)