```
[1]
     <PROGRAM>
                             \rightarrow <IDENT><ENV><DATA><PROC>
[2]
     <IDENT>
                                identification division <END INST> program-id.
                                 ID<END INST> author. WORDS<END INST>
                                 date-written. WORDS <END INST>
     <WORDS>
[3]
                                 ID < WORDS LR>
[4]
     <WORDS LR>
                                ID <WORDS LR>
[5]
[6]
     <END INST>
                                \cdot n
                                 environment division
<br/>
\langle {\rm END\_INST} \rangle configuration
[7]
     <ENV>
                                 section<END_INST> source-computer.
                                 WORDS<END INST> object-computer.
                                 WORDS<END INST>
                                 data division<END INST> working-storage
[8]
     <DATA>
                                 section<END INST> <VAR LIST>
[9]
     <VAR LIST>
                                 <VAR DECL> <VAR LIST>
[10]
     <VAR DECL>
                             \rightarrow <LEVEL> ID pic IMAGE <VAR DECL TAIL>
[11]
[12]
    <VAR_DECL_TAIL>
                                value INTEGER<END INST>
[13]
                                 <END INST>
[14]
     <LEVEL>
                                INTEGER
     <PROC>
                                 procedure division<END INST> ID
[15]
                                 section<END INST> <LABELS> end program ID.
                                 <LABEL><END_INST> <INSTRUCTION_LIST>
[16]
    <LABELS>
                                 <LABELS LR>
[17]
                                 <LABEL><END INST> <INSTRUCTION LIST>
     <LABELS LR>
                                 <LABELS LR>
[18]
                                \varepsilon
[19]
     <LABEL>
                                ID
[20]
     <INSTRUCTION_LIST>
                                <INSTRUCTION> <INSTRUCTION LIST>
[21]
[22]
     <INSTRUCTION>
                                <ASSIGNATION>
[23]
                             \rightarrow <IF>
[24]
                             \rightarrow <CALL>
[25]
                             \rightarrow <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>
                                 subtract < EXPRESSION > from \ ID < END \quad INST >
[31]
[32]
                                multiply <ASSIGN END><END INST>
[33]
                                 divide <ASSIGN_END><END_INST>
[34]
    <ASSIGN END>
                                 <EXPRESSION>,<EXPRESSION> giving ID
[35]
     <EXPRESSION>

ightarrow <EXP AND> <EXPRESSION LR>
[36]
    <EXPRESSION LR>
                             \rightarrow or <EXP AND> <EXPRESSION LR>
[37]
                             \rightarrow
    \langle \text{EXP\_AND} \rangle

ightarrow <EXP EQUAL> <EXP AND LR>
[38]
[39]
    <EXP AND LR>
                                 and <EXP EQUAL> <EXP AND LR>
```

```
[40]
[41]
    <EXP EQUAL>
                            \rightarrow <EXP ADD> <EXP EQUAL LR>
[42]
    <EXP EQUAL LR>
                            \rightarrow = <EXP ADD> <EXP EQUAL LR>
[43]

ightarrow < <EXP ADD> <EXP EQUAL LR>
                            \rightarrow > <EXP ADD> <EXP EQUAL LR>
[44]
[45]
                               <= <EXP_ADD> <EXP_EQUAL_LR>
                               >= <EXP ADD> <EXP EQUAL LR>
[46]
[47]
    <EXP ADD>
                               <EXP MULT> <EXP ADD LR>
[48]
    <EXP\_ADD\_LR>
                               + <EXP MULT> <EXP ADD LR>
[49]
[50]
                            \rightarrow - <EXP MULT> <EXP ADD LR>
[51]
[52]
    <EXP_NOT> <EXP_MULT_LR>
                            [53]
    <EXP_MULT_LR>
                               / < EXP NOT > < EXP MULT LR >
[54]
                            \rightarrow
[55]
[56]
    <EXP_NOT>
                            \rightarrow -<EXP NOT>
[57]
                               not <EXP_NOT>
[58]
                               <EXP PARENTHESIS>
[59]
    <EXP PARENTHESIS>
                            \rightarrow (\langle EXPRESSION \rangle)
[60]
                               <EXP TERM>
    <\!\!\mathrm{EXP\_TERM}\!\!>
[61]
                               ID
[62]
                               INTEGER
                            \rightarrow
[63]
                               true
[64]
                               false
[65]
    \langle IF \rangle
                               if <EXPRESSION> then <INSTRUCTION_LIST>
                                <IF END>
[66]
    <IF END>
                                else <INSTRUCTION LIST> end-if
[67]
                                end-if
[68]
    <CALL>
                                perform ID < CALL_TAIL>
                                until <EXPRESSION><END INST>
[69]
    <CALL_TAIL>
                               <END INST>
[70]
[71]
    <READ>
                                accept ID<END INST>
[72]
                               display <WRITE TAIL>
    <WRITE>
[73]
    <WRITE_TAIL>
                               <EXPRESSION><END_INST>
[74]
                            → STRING<END INST>
```

Variable	First^1	$Follow^1$
<program></program>	identification	
<IDENT $>$	identification	first(ENV)
<WORDS $>$	ID	$first(END_INST)$
$<$ WORDS_LR $>$	ID, ε	follow(WORDS)

<end_inst></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></env>	environment	first(DATA)
<data></data>	data	first(PROC)
<var list=""></var>	INTEGER, ε	follow(DATA)
<var decl=""></var>	INTEGER	first(VAR LIST)
<var decl="" tail=""></var>	value, .	follow(VAR DECL)
<level></level>	INTEGER	ID
<proc></proc>	procedure	follow(PROGRAM)
<labels></labels>	ID	end
<labels lr=""></labels>	ID, ε	follow(LABELS)
<label></label>	ID	first(END_INST)
<instruction list=""></instruction>	move, compute, add, substract,	first(LABELS LR)
<instruction></instruction>	multiply, divide, if, perform, accept, display, stop, ε move, compute, add, substract, multiply, divide, if, perform,	$first(INSTRUCTION_LIST)$
<assignation></assignation>	accept, display, stop move, compute, add, substract, multiply, divide	${\rm follow}({\rm INSTRUCTION})$
<assign end=""></assign>	-, not, (, ID, INTEGER, true, false	first(END INST)
<expression></expression>	-, not, (, ID, INTEGER, true, false	to, first(END_INST), from, , , giving,) then
$<$ EXPRESSION_LR $>$	or, ε	follow(EXPRESSION)
<exp_and></exp_and>	-, not, (, ID, INTEGER, true, false	$first(EXPRESSION_LR)$
<exp_and_lr></exp_and_lr>	and, ε	$follow(EXP_AND)$
$<$ EXP $_$ EQUAL $>$	-, not, (, ID, INTEGER, true, false	$first(EXP_AND_LR)$
		$follow(EXP_EQUAL)$
<exp_add></exp_add>	-, not, (, ID, INTEGER, true, false	$first(EXP_EQUAL_LR)$
<exp_add_lr></exp_add_lr>	$+, -, \varepsilon$	follow(EXP_ADD)
<exp_mult></exp_mult>	-, not, (, ID, INTEGER, true, false	first(EXP_ADD_LR)
<exp_mult_lr></exp_mult_lr>	*, /, ε	follow(EXP_MULT)
<exp_not></exp_not>	-, not, (, ID, INTEGER, true, false	first(EXP_MULT_LR)
<exp_parenthesis></exp_parenthesis>	(, ID, INTEGER, true, false	follow(EXP_NOT) follow(EXP_PARENTHESIS)
<exp_term> <if></if></exp_term>	ID, INTEGER, true, false if	follow(INSTRUCTION)
<if end=""></if>	else, end-if	follow(IF)
<call></call>	perform	follow(INSTRUCTION)
<call_tail></call_tail>	until, .	follow(CALL)
	, , -	- ()

<read> <write> <write_tail></write_tail></write></read>	accept display STRING, -, not, (, ID, INTEGER, true, false	follow(INSTRUCTION) follow(INSTRUCTION) follow(WRITE)