Language Specification:

Language Definition:

1.1 Alphabet:

1.1.a. Upper (A-Z) and lower case letters (a-z) of the English alphabet

b. Underline character '_';

c. Decimal digits (0-9);

Lexic:

a.Special symbols, representing:

- operators + - * / ++ := < <= = >=
- separators [ ] { } : ; space
- reserved words:

array char const do else if int of program read

then var while write swap of type size

b.identifiers

-a sequence of letters and digits, such that the first character is a letter; the rule is:

identifier ::= letter | letter{letter}{digit}

letter ::= "A" | "B" | . ..| "Z"

digit ::= "0" | "1" |...| "9"

c.constants

1.integer - rule:

noconst:=+no|-no|no

no:=digit{no}

2.character

character:='letter'|'digit'

Array

size:=number of identifiers

type:= INT | CHAR |

array:= ARRAY[size] OF type

Comments:

1.Single-line Comments:

“ // “ to start a comment that continues to the end of the line

2. Multi-line Comments:

“ /* ” to begin a comment and “ */ “ to end it. This allows for comments that span multiple lines

Syntax:

The words - predefined tokens are specified between " and ":

Sintactical rules:

program ::= "VAR" decllist ";" cmpdstmt "."

decllist ::= declaration | declaration ";" decllist

declaration ::= identifier ":" "INT"

cmpdstmt ::= "BEGIN" stmtlist "END"

stmtlist ::= stmt | stmt ";" stmtlist

stmt ::= simplstmt | structstmt

simplstmt ::= assignstmt | iostmt

assignstmt ::= identifier ":=" expression

expression ::= expression "+" term | expression "-" term | term

term ::= term "*" factor | term "/" factor | factor

factor ::= "(" expression ")" | identifier | integer

iostmt ::= "READ" "(" identifier ")" | "WRITE" "(" expression ")"

structstmt ::= ifstmt | whilestmt | forstmt

ifstmt ::= "IF" condition "THEN" stmt ["ELSE" stmt]

whilestmt ::= "WHILE" condition "DO" stmt

forstmt ::= “FOR” assignstmt “TO” expression “DO” stmt //incrstmt

incrstmt ::= identifier + 1 “++”

condition ::= expression RELATION expression

RELATION ::= "<" | "<=" | "=" | "<>" | ">=" | ">"

```
swapstmt ::= "SWAP" "(" identifier ", " identifier ")"        //this swaps the two identifiers
```

b)PROGRAM: (Bubble Sort)

```
VAR
         arr: ARRAY[5] OF INT;
         i: INT;
         j: INT;
         temp: INT;
        n: INT;

BEGIN
         arr[0] := 5;
         arr[1] :=7;
         arr[2] := 18;
         arr[3] := 4;
         arr[4] := 14;
         n := 5;
        FOR i := 0 TO n - 1 DO
                FOR j := 0 TO n - i - 2 DO
                        IF arr[j] > arr[j + 1] THEN
                                SWAP(arr[j], arr[j + 1]);
                        END;
                END;
        END;
/*
        The array will be sorted
        arr[] = [4, 5, 7, 14, 18];
*/
```

END.