Oct. 4, 2003 **ICS 313-02 (031)** Dr. Mamdouh Najjar

Fundamentals of Programming Languages

Programming Assignment # 1
Interpreter for a Simple Language (*SIM v.1.0*)
Due Nov. 1, 2003

A. **Given *SIM v.1.0* syntax described in BNF rules as follows:**

<program> → Start_Program <program_name> ; < blocks> End_Program <program_name>;
<program_name> → <id>
<blocks> → <datablock> <statblock>
<datablock> → Begin_Define <datatypes> End_Define;
<statblock> → Begin_Block; <stmt_list> End_Block ;
<stmt_list> → <stmt> | <stmt> ; <stmt_list>
<stmt> → <assign> | <read> | <write>
<assign> → <var> := <expr>
<read> → Read <varlist>;
<write> → Write <varlist>;
<expr> → <expr> + <term> | <expr> - <term> | <term>
<term> → <term> * <factor> | <term> / <factor> | <factor>
<factor> → (<expression>) | <var> | <sign_integer>
<var> → <id>
<id> → <char> | <char> <id>
<char> → a | b | c | … | z
<varlist> → <id> | <id> , <varlist>
<datatypes> → <datatype> | <datatype> <datatypes>
<datatype> → <vartype> | <consttype>
<vartype> → Var <varlist> : <varlast> ;
<varlast> → <varkind> | Array [<sign_integer> .. <sign_integer>] Of <varkind>;
<varkind> → Integer | Real | Boolean | Char
<consttype> → Constant <id> = <value>;
<value> → '<string>' | <sign_integer> | <real>
<string> → <char> | <char> <string>
<real> → <sign_integer> . <integer>
<sign_integer> → <sign> <integer>
<sign> → + | - | NULL
<integer> → <digit> | <digit> <integer>
<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

B. **Your Program**

Your program should read a program written in the ***SimLan V2*** and executes its statements if syntacticaly correct. In case of a syntax error, your program should print the erronous line then an error message indicating the type of error. Error types are:

- unmatched blocks
- type error
- invalid expression

- invalid statement
- no value assigned to variable XX
- Input error

This grammer will allow the generation of programs like:

```
NTWO START;
        DEFINE
                VAR A, B      :       INTEGER;
                VAR C, D      :       REAL;
                VAR E         :       CHAR;
                VAR F         :       ARRAY [-2..4] OF CHAR;
                TYPE TESTREC          RECORD
                                      RI      :       BOOLEAN;
                                      RII     :       REAL;
                                      END;
                CONSTANT  CI          =       3.1415;
                CONSTANT  CII         =       8;
        END;
                BEGINB;
                READ A, B;
                F[1]    := “I”
                F[2]    := “LIKE”
                F[3]    := “ICS313”
                A       := CI * (B + CII * A);
                B       :=      CII;
                D       :=      CI * CII
                E       :=      F[3];
                C       :=      CI;
                PRINT A,B,C,D, E, CI, CII;
                ENDB;
```

If the input to this program is:
4 10
then the output of the program is:
131.943 8 3.1415 25.132 ICS313 3.142 8

Important notes:

1- Your program must read from an input file and print to an output file.

2- The source file must include the following:

- your name, ID number
- Course title, number, and section number
- the statement of the problem
- a brief summary of your interpreter implementation method
- a dictionary of all global variables (name, type, and usage)
- for each procedure or function include the following:
    - Usage of the procedure or function
    - Meanings of the input variables
    - Meanings of the output variables

3- Late assignments will not be accepted.