Alphabet:

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

b.Decimal digits (0-9);

c.Underline character: "_"

Lexic:

a.Special symbols, representing:

-operators: + - * / = < > <= >= !=

-separators: space , ( ) [ ]

-reserved words : INCEPUT, SFARSIT, DACA, ATUNCI,ALTFEL, CITESTE, SCRIE, CAT_TIMP, EXECUTA,INT, BOOL,CHAR, SIR, CONST, VAR

b.indetifiers:

-a sequence of letters and digits, first character is always a letter

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

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

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

c.constants

1.integer

nrconstant:="+"nr|"-"nr|nr

nr:=nonzero_digit{nr}

nonzero_digit::= "1" |...| "9"

2.character

character:='letter'|'digit'

constchar:=" ' " character " ' "

3.string

conststring=" " " string " " " | ""

string:=char{string}

char:=letter|digit

2.Syntax:

program ::= "INCEPUT" "{"decllist"}" ";" "{"cmpdstmt "}" "SFARSIT"

decllist ::= declaration | declaration "," decllist

declaration ::= type IDENTIFIER

type1 ::= "BOOL" | "CHAR" | "INT"

arraydecl::="SIR" "(" nr ")" "DE" type1

type ::= type1|arraydecl

cmpdstmt ::= "(" stmtlist ")"

stmtlist ::= stmt | stmt "," stmtlist

stmt ::= simplstmt | structstmt

simplstmt ::= assignstmt | iostmt

iostmt ::= "CITESTE" IDENTIFIER | "SCRIE" (IDENTIFIER | CONSTANT)

assignstmt ::= IDENTIFIER "=" expression

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

term ::= term ("*"|"/"|"%") factor | factor

factor ::= expression | IDENTIFIER | constant

ifstmt ::= "DACA" condition "ATUNCI" stmt ["ALTFEL" stmt]

whilestmt ::= "CAT_TIMP" condition "EXECUTA" stmt

condition ::= expression RELATION expression

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