Lexic.txt

Alphabet:

-Upper and lower case letters of the English alphabet

-decimal digits 0-9

-special symbols:

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

separators : - space \n tab

reserved words: fnct daca nuECazu exista iaCateUna sight raspuns vorbe numar vorba

-identifiers:

a sequence or at least one letter, such that the first character is a letters

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

identifier ::= letter | {letter}

-constants

integer:

constnumar := "numar"

nonzerodigit := "1" | ... |"9"

numar := nonzerodigit["0"|nonzerodigit]

character

constchar := "vorbe"

vorba := letter |digit

string

vorbe := vorba {vorbe}

token.in

Keywords

fnct

daca

nuECazu

exista

iaCateUna

sight

raspuns

vorbe

vorba

numar

citeste

Operators

+

-

=

*

/

<

>

<=

>=

!=

Separators

:

-

\n

Syntax.in

Syntactical rules:

relation ::= "<" | "<=" | "=" | "!=" | ">=" | ">" | "si" | "sau"

term ::= identifier

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

condition ::= expression RELATION expression

whilestmt ::= "catTimp" condition ":" "fa" stmt

ifstmt ::= "daca" condition ":" "nuECazu" stmt ["nuECazu" stmt]

structstmt ::= cmpdstmt | ifstmt | whilestmt

iostmt ::= "citeste" | "sight" "(" IDENTIFIER ")"

forstm ::= "iaCateUna" array ":"

stmt ::= ifstmt | whilestmt | forstm | iostmt | assignstmt

assignstmt ::= identifier "=" expression

stmtlist ::= stmt "\n" | stmt "\n" stmtlist

type1 ::= "numar" | "vorbe" | "vorba"

array ::= identifier "[]"

arraydecl ::= type1 identifier "[]"

declaration ::= type1 identifier

function ::= "fnct - " nume ":" identifierList "\n" stmtlist |stmt

program ::= function | declaration