1. Specificarea minilimbajului de programare

Minilimbajul este un subset al limbajului C++. Contine tipuri de date simple (int, double) si compuse (int[]) si permite instructiuni de atribuire, intrare/iesire, selectie(conditionate) si de ciclare

Clasificarea atomilor

1. Identificatori

ID ::= litera (litera|cifra|”\_”) {0,7} -> restrictive 1.a.

litera ::= “A”|”B”|…|”Z”|”a”|”b”|…|”z”

cifra ::= ”0”|“1”|”2”|“3”|”4”|”5”|”6”|”7”|”8”|”9”

2. Constante

CONST\_STRING ::= “”” {cifra | litera | caracter\_special} “””

cifra ::= ”0”|“1”|”2”|“3”|”4”|”5”|”6”|”7”|”8”|”9”

litera ::= “a”|”b”|”c”| … | “z” |”A”|”B”|…|”Z”

caracter\_special ::= “~”|”!”|”@”|”#”|”$”|”%”|”^”|”&”|” “ |”\*“|”(“|”)”|”\_”|”-”|”+”|”=”|

“`”|”[”|”{“|”]”|”}”|”\”|”|”|”;”|”:”|”’”|””””|”,”|”<”|”.”|”>”|”?”|”/”

CONST\_POZITIV ::= cifra\_nenula (cifra\_nenula|”0”)\* [ “.” (cifra\_nenula|”0”)\* cifra\_nenula ]

CONST ::= “0” | ([“+”|”-”] cifra\_nenula (cifra\_nenula|”0”)\* [ “.” (cifra\_nenula|”0”)\* cifra\_nenula ])

cifra\_nenula ::= “1”|”2”|“3”|”4”|”5”|”6”|”7”|”8”|”9”

3. Cuvinte rezervate

“int”, “double”, “void”, “main”, “cout”, “cin”, “while”, “if”, “else”, “M\_PI”, “endl”

4. Operatori

“+”, “-”, “\*”, “<<”, “>>”, “=”, “!=”, “> ”, ”<”, “<=”, “>=”, “==”, “[”, “]”

5. Separatori

“(”, “)”, “{”, “}”, “,”, “;”, “ ”, “ “ ”

Specificarea minilimbajului de programare folosind EBNF

program ::= “int” “main” “(” “)” “{” [lista\_decl] [lista\_instr] “}”

lista\_decl ::= dec “;” {dec “;”}

dec ::= tip lista\_def

tip ::= (“int” | “double”) [ “[” CONST\_POZITIV|ID “]” ]

lista\_def ::= definire {“,” definire}

definire ::= ID [ “=” CONST ]

lista\_instr ::=instr {“;” instr}

instr ::= atribuire | instr\_if | instr\_loop | intrare | iesire

atribuire ::= ID “=” variabila { operator\_ aritm variabila}

variabila ::= ID | CONST | variabila\_compusa

variabila\_compusa ::= ID “[” (CONST\_POZITIV | ID) “]”

operator\_aritm ::= “\*”|”-”|”+”

instr\_if ::= “if” ”(” conditie “)” “{” lista\_instr “}” [“else” “{” lista\_instr “}” ]

conditie ::= variabila operator\_rel variabila

operator\_rel ::= “!=”|“==”|“> ”|”<”|“<=”|“>=”

instr\_loop ::= “while” “(” conditie “)” “{” lista\_instr ”}”

intrare ::= “cout” scriere {scriere}

scriere ::= “<<” ( variabila | “endl” | CONST\_STRING )

iesire ::= “cin” “>>” ID

1. Se cer textele sursa a 3 mini-programe -> lab1.cpp

3.Se cer textele sursa a 2 programe care contin erori

A. doua erori in limbajul original (inclusive MLP)

Int main()

{

double nums[100], sum = 0;

int n, i = 0 //->err: fara ; dupa decl

cout << "n="; cin>>n;

while (i < ) // ->err : fara a doua variabila din conditie

{

cout << "nums[" << i << "]=";

cin >> nums[i];

sum = sum + nums[i];

i = i + 1;

}

cout << "sum: " << sum << endl;

}

A. doua erori in MPL (fara limbajul original)

Int main()

{

double arrayOfNums[100], sum = 0; //err -> ID cu lungime //mai mare de 8 caractere

int n, i = 0;

cout << "n="; cin>>n;

for( i = 0; i < n ; i++) // ->err : nu exista for in MLP

{

cout << "nums[" << i << "]=";

cin >> nums[i];

sum = sum + nums[i];

i = i + 1;

}

cout << "sum: " << sum << endl;

}