# Tema 1

## Specificarea minilimbajului de programare (MLP).

Limbajul ales: C++

ID 🡪 <litera> | <litera> ID | <litera> <cifra> ID

<litera> 🡪 “A” | “B”|…|”Z”|”a”|”b”|…|”z”

CONST 🡪 <nr\_intreg> | <nr\_intreg> “.” <nr\_intreg>

<nr\_intreg> 🡪<cifra> | <cifra> <nr\_intreg>

<cifra > 🡪 “0” | “1” | … |”9”

CONST 🡪 R

<program> 🡪 “start” <lista\_instructiuni> “end”

<citire> 🡪 “read” <stream\_citire>

<stream\_citire> 🡪 “>>” ID | “>>” ID <stream\_citire> | “>>” ID “.” ID

<scriere> 🡪 “write” <stream\_scriere>

<stream\_scriere> 🡪 “<<” ID | “<<” <stream\_scriere> | “<<” ID “.” ID

<lista\_instructiuni> 🡪 <instr> <lista\_instructiuni> | <instr>

<instr> 🡪 <atribuire> “;”| <citire>”;” | <scriere> “;” | <conditionala>

| <repetititva> | <declaratie>”;”

<atribuire> 🡪 ID “=” <expresie>

<expresie> 🡪 ID | CONST | ID <operatie> ID | ID <operatie> CONST | ID <operatie> <expresie> | CONST <operatie> <expresie> | ID “.” ID

<operatie> 🡪 “+” | “-” | “/” | “%”|”\*”

<conditionala> 🡪 “if” “(”<conditie> ”)” ”{” <lista\_instructiuni> ”}”

| “if” “(”<conditie> ”)” ”{” <lista\_instructiuni> ”}” “else” “{“<lista\_instructiuni>”}”

<conditie> 🡪 <expresie\_logica> | <expresie\_logica> <operator\_logic> <conditie>

<expresie\_logica> 🡪 ID <comparator>ID | ID <comparator> CONST

<comparator> 🡪 “<” | “>” | “==” | “!=”

<operator\_logic> 🡪 “&&” | “||”

<repetitiva> 🡪 “while” “(“<expresie\_logica> “)” “{” <lista\_instructiuni>“}”

<tip\_de\_data\_predef> 🡪 “struct” ID “{” <declaratie> “}” “;”

<lista\_declaratii> 🡪 ID | ID “,” <lista\_declaratii>

<declaratie> 🡪 <tip\_data> <lista\_declaratii>

<tip\_data> 🡪 “int” | “bool” | “float” | <tip\_de\_data\_predef>

## Se cer textele sursa a 3 mini-programme

### Calculeaza perimetrul si aria cercului de o raza data

start

float pi ;

int perimetru ,aria;

int raza ;

read>>raza;

perimetru=2\*pi\*raza ;

write<<perimetru;

aria=pi\*raza\*raza;

write<<aria;

end

### Determina cmmdc a 2 nr naturale

start

int a,b;

read>>a>>b;

if(a==0){

write<<b;

}

else{

if(b==0){

write<<a;

}

else{

while(a!=b){

if(a>b){

a=a-b;

}

else{

b=b-a;

}

}

write<<a;

}

}

end

### Calculeaza suma a n numere citite de la tastatura

start

int numar, n,suma;

read>>n;

suma=0;

while(n!=0){

read>>numar;

suma=suma+numar;

n=n-1;

}

write<<suma;

end

## Se cer textele sursa a doua programe care contin erori conform MLP-ului definit:

### Unul dintre programe contine doua erori care sunt in acelasi timp erori in limbajul original (pentru care MLP defineste un subset)

Suma a doua numere.

start

int a,b,suma;

read>>a>>b;

suma=a+b - lipseste punct virgula

write<suma; - lipseste un <

end

### Al doilea program contine doua erori conform MLP, dar care nu sunt erori in limbajul original. Se cere ca acesta sa fie compilat si executat in limbajul original ales.

Suma primelor n numere

Int main(){ -- eu am start si end

int n, suma;

cin>>n ;

suma=0 ;

while(n !=0){

suma+=n ; --eroare in limbajul meu

n-- ; --eroare in limbajul meu

}

cout<<suma ;

}

## Program ce identifica atomii

Atomi: ID, CONST, “start” , “end”, “read”, “>>”, “.”, “write”, “<<”, “;”, “=”, “+” , “-” , “/” , “%”, ”\*”, “if”, “(”, ”)” , ”{”, ”}” , “else”, “<” , “>” , “==” , “!=”, “&&”, “||”, “while” ,“struct”, “int”, “bool”, “float”, “,”

# Tema lab2 in clasa

Produsul primelor n numere

int main(){

int p = 1; -- correct in c++; in limbajul lui se poate doar int p;

int n;

cin>>n; --corect in c++, in lbj lui este read(n)

while (n > 0){ -- paranteze corecte in c++ , la el nu exista paranteze

p \*= n; -- correct in c++, in lbj lui se poate doar p = p \* n;

n = n – 1;

}

cout<<p;

}

# Tema 2

ac