**LFTC LAB 1**

1. **Specificarea minilimbajului de programare (MLP).**

<program> := <antet\_program> <functie>

<antet\_program>:=<tip>ID | <lista decl fct>

<functie> := <antet\_functie> <corp>

<antet\_functie> := <tip> ID (<lista\_decl>) | <tip> ID ()

<tip> := int | float | string

<lista\_decl\_fct> := <declarare\_fct> , <lista\_decl\_fct> | <declarare\_fct>

<declarare\_fct> := <tip> ID

<corp> := { <instr\_compusa> }

<instr\_compusa> := <instr> <instr\_compusa> | <instr>

<instr> := <declarare> | <atribuire> | <instr\_citire> | <instr\_afisare> | <instr\_while> | <instr\_if>

<declarare> := <tip> ID ; (\n)\* | <tip> <atribuire> (\n)\*

<atribuire> := ID := <expr\_aritmetica> ;

<instr\_citire> := citește ( ID ) ;

<instr\_afisare> := scrie ( <expr\_aritmetica> ) ; (\n)\*

<expr\_aritmetica> := <expr\_aritmetica> <op\_artimetic> <expr\_aritmetica> | ID | CONST

<op\_artimetic> := + | - | \* | / | %

<instr\_while> ::= repetă ( <conditie> ) <corp> <conditie> ::= <expr\_aritmetica> <op\_relational> <expr\_aritmetica> | <expr\_aritmetica>

<op\_relational> ::= != | == | < | > | <= | >=

<instr\_if> ::= dacă ( <conditie> ) <corp> altfel <corp>

ID ::= (\_ | a | b | ... | z | A | B | ... | Z)(\_ | a | b | ... | z | A | B | ... | Z | 0 | 1 | ... | 9){0,249}

CONST ::= <const\_int> | <const\_float> | <const\_string>

<const\_int> ::= (+ | -)?(0 | 1 | ... | 9)+

<const\_float> ::= (+ | -)?(0 | 1 | ... | 9)+ | (+ | -)?(0 | 1 | ... | 9)+\.(0 | 1 | ... | 9)\*

<const\_string> ::= ".\*"

sau

ID ::= ^[\_a-zA-Z]([\_a-zA-Z0-9]){0,249}$

CONST ::= <const\_int> | <const\_float> | <const\_string>

<const\_int> ::= ^[+-]?[0-9]+$

<const\_float> ::= ^[+-]?[0-9]+(\.[0-9]\*)?$

<const\_string> ::= ^".\*"$

1. **se cer textele sursa a 3 mini-programe**

calculeaza perimetrul si aria cercului de o raza data data

int main() {

float r;

citeste(r);

float pi = 3.14;

//calculez perimetrul

float perimetru = 2 \* pi \* r;

scrie(perimetru);

//calculez aria

float aria = pi \* r \* r;

scrie(aria);

}

determina cmmdc a 2 nr naturale

int main() {

int a;

citeste(a);

int b;

citeste(b);

repeta(a!b){

daca(a>b) a = a – b;

altfel

b = b – a;

}

Scrie(a);

}

calculeaza suma a n numere citite de la tastatura

int main() {

int s = 0;

int n;

int x;

citeste (n);

while (n > 0)

{

citeste (x);

s = s + x;

n = n - 1;

}

scrie (s);

}

1. **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)

int main() {

stringg a; //eroare

int b //eroare

citeste(b);

scrie(a+1);

}

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.

Int main() {

Int a;

Int I;

float b;

citeste(a);

citeste(b);

for(i=0;i<=a;i++) { //eroare nu am definit for

scrie(b);

}

return 0; //eroare pt ca nu e definit return

}