Tema 1

EBNF:

Program = Lista\_importuri Lista\_declarari Main .

Lista\_importuri = import {import} .

Import = “include<” Biblioteca “>” .

Biblioteca = “iostream” .

Biblioteca = “cmath” .

Lista\_declarari = Declarare {Declarare Declarare\_struct} .

Declarare = Tip\_data Lista\_variabile “;” .

Declarare = “const double pi = 3.14;” .

Declarare\_struct = “struct MyType{” Declarare “};” .

Tip\_data = “int” .

Tip\_data = “double” .

Tip\_data = “typedef struct MyType {” Lista\_variabile “}” .

Lista\_variabile = Id { “,” Id } .

Id = Litera {Litera} .

Litera = “A” | “B” | … | “Z” {Litera} .

Main = “int main() {“ Lista\_instructiuni ” return;” } .

Lista\_instructiuni = Instructiune {Instructiune} .

Instructiune = Atribuire .

Instructiune = Intrare .

Instructiune = Iesire .

Instructiune = Conditionare .

Instructiune = Ciclare .

Atribuire = Id “=” Expresie “;” .

Expresie = Termen { “+“ | “-“ Termen} .

Termen = Factor { “\*” | “/” | “%” Factor} .

Factor = “(“ Expresie “)” .

Factor = Id .

Id = Litera {Litera}.

Litera = “a” | “b” … | “z” .

Factor = Numar .

Numar = CifraN {Cifra} .

CifraN = “1” | “2” | … “9” .

Cifra = “0” | “1” | “2” | … | “9” .

Intrare = “cin>>” Id “;” .

Iesire = “cout<<” Id [“<<’ ‘”] “;”.

Conditionare = “if(“ Expresie Operator\_comparatie Expresie “) {” Lista\_instructiuni” “}” {“else {“ Lista\_instructiuni ”}”} .

Operator\_comparatie = “==” | “>” | “<” | “<=” | “>=” | “!=” .

Ciclare = “while(” Expresie Operator\_comparatie Expresie “){“ Lista\_instructiuni” “}” .

2.

include <iostream>

int r;

double aria;

double perimetru;

double pi;

int main()

{

p = 3.14;

cin >> r;

perimetru = 2 \* pi \* r;

aria = pi \* r \* r;

cout << perimetru << " ";

cout << aria;

}

include <iostream>

int x;

int y;

int main()

{

cin >> x;

cin >> y;

while (x != y)

{

if (x > y)

{

x = x - y;

}

else

{

y = y - x;

}

}

cout << x;

return;

}

include <iostream>

int n;

int x;

int suma;

int main()

{

cin >> n;

while (n != 0)

{

cin >> x;

suma = suma + x;

n--;

}

cout << suma;

return;

}

3.

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

Erori: cinn >> n; x-- -;

#include <iostream>

using namespace std;

int n, x, suma;

int main()

{

x-- - ;

cinn >> n;

while (n != 0)

{

cin >> x;

suma = suma + x;

n = n - 1;

}

cout << suma;

return 0;

}

2. 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.

Erori: cin >> x >>y, lipseste return 0;

#include <iostream>

using namespace std;

int x, y;

int main()

{

cin >> x >> y;

while (x != y)

{

if (x > y)

{

x = x - y;

}

else

{

y = y - x;

}

}

cout << x;

}

MLP Maria, structure sintactice:

* Nu are if simplu, doar cu else
* Nu are for
* Nu are initializare la declarare
* Nu are do/while
* Nu are clase