Documentatie lab 1 LFTC

Specificarea minilimbajului de programare (MLP)

Extended Backus-Naur Form

program = program_header “{“ lista_decl “return” CONST ”;” “}“.
program_header = “#include” “<iostream>” tip “main” “(“ “)”.
lista_decl = decl [“;”] { decl [“;“] }.
decl = variabila_decl | attr_decl | if_instr | while_instr | cin_instr | cout_instr.
variabila_decl = [“const”] tip ID [“=” expr] { “,” tip ID [“=” expr] } “;”.
tip = “int” | “double”.
tip_special = “struct” “{“ variabila_decl “}” “TipSpecial” “;”.
attr_decl = ID “=“ expr “;”.
while_instr = “while” “(“ expr “)” bloc_instr.
cin_instr = “std::cin” { “>>“ ID } “;”.
cout_instr = “std::cout” { “<<“ (ID | ([““”] CONST [““”])) } “;”.
if_instr = "if" "(" expr ")" bloc_instr ["else" block_instr].
block_instr = decl | ("{" lista_decl "}").
expr = expr_term { operator expr_term }.
expr_term = "(" expr ")" | ID | CONST.
operator = "+" | "-" | "*" | "/" | "%" | "&&" | "||" | "==" | "!=" | "<" | "<=" | ">" | ">=".
switch_instr = “switch” “(“ expr “)” “{“ block_switch “}”.
block_switch = { caz_switch }.
caz_switch = ((“case” expr) | “default”) “:” [“{“] expr [“break” “;”] [“}”].

Restricții

ID - să înceapă cu o literă și să fie format doar din litere și cifre.
CONST - poate fi orice caracter fără apostrof simplu și dublu.

Textele sursă pentru cele 3 mini-programe corecte

Calculează perimetrul și aria cercului de o rază dată

```
#include <iostream>

int main()
{
    const double PI = 3.14159265358979323846;
    double raza = 0.0;
```

```
    std::cin >> raza;

    double aria = PI * raza * raza;
    double perimetru = 2 * PI * raza;
    std::cout << aria << " " << perimetru;

    return 0;
}
```

Determină cel mai mare divizor comun a 2 numere naturale

```
#include <iostream>

int main()
{
    int a = 0, b = 0, rest = 0;
    std::cin >> a >> b;

    while (b != 0)
    {
        int rest = a % b;
        a = b;
        b = rest;
    }
    std::cout << a;

    return 0;
}
```

Calculează suma a n numere citite de la tastatură

```
#include <iostream>

int main()
{
    int i = 0, numar = 0, suma = 0, n = 0;
    std::cin >> n;

    while (i < n)
    {
        std::cin >> numar;
        suma = suma + numar;
        i = i + 1;
    }
    std::cout << suma;

    return 0;
}
```

Textele sursă pentru cele 2 mini-programe greșite

Două erori care sunt în același timp erori în limbajul original (pentru care MLP definește un subset)

```
using <iostream>

int main()
{
    const int 1unu1 = 1;
```

```
6
7       return 0;
8   }
```

Două erori conform MLP, dar care nu sunt erori în limbajul original

```
1   #include <iostream>
2
3   int main()
4   {
5       if (1 == 1);;;
6
7       int a = 0; ++a;
8
9       return 0;
10  }
```