**LFTC – MLP Simplificat**

Laborator6

1.

- Tipuri de Date:

* int: Tip de date utilizat pentru a reprezenta numere întregi, curpinse între -2.147.483.648 și 2.147.483.647 (pe 32 de biți)

- Instrucțiuni:

* **Instrucțiune de atribuire**: variabilă = expresie;
* **Instrucțiune de intrare/ieșire**: cin >> variabilă; cout << expresie;

**Analizator Lexical**:

Simboluri utilizate:

1. Operatori:

- Aritmetici: “+”, “-”, “\*”

- Atribuire: “=”

2. Cuvinte cheie:

- Tipuri de date permise: “int”

- Instructiuni de intrare/iesire: “cin >>”, “cout <<”

- Start program: “main”

- Linie nouă: “endl”

3. Identificatori:

- identificator = litera { litera | cifra | “\_” }

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

- cifra = “0” | “1” | … | “9”

4. Delimitatori: “;”, “,”, “(“, “)”, “{“, “}”

EBNF:

program = "void main() {" {declarații\_var} {instrucțiuni} "}".

ID = literă {literă | cifră | “\_”}.

declaratii\_var = “int” (ID | instr\_atribuire) “;”.

instrucțiuni = (instr\_intrare | instr\_ieșire | instr\_atribuire) .

instr\_atribuire = ID ”=” expresie ”;”.

instr\_intrare = ”cin >>” ID {”>>” ID} ”;”.

instr\_ieșire = “cout <<” (expresie | “endl”) {“<<” (expresie | “endl”)} ”;”.

expresie = CONST | ID | operație.

operație = (ID | CONST) operator\_artimetic ((ID | CONST) | {operație}).

CONST = [“-“] cifră\_nenulă {cifră}.

literă = “a” | “b” | … | “z” | “A” | “B” | … | “Z”.

cifră = “0” | “1” | … | “9”.

cifră\_nenulă = “1” | … | “9”.

operator\_aritmetic = “+” | “-” | “\*”.

|  |  |
| --- | --- |
| **Atom** | CodAtom |
| ID | 0 |
| CONST | 1 |
| void | 2 |
| main | 3 |
| ( | 4 |
| ) | 5 |
| { | 6 |
| } | 7 |
| ; | 8 |
| , | 9 |
| int | 10 |
| = | 11 |
| cin | 12 |
| >> | 13 |
| cout | 14 |
| << | 15 |
| endl | 16 |
| + | 17 |
| - | 18 |
| \* | 19 |