1. Identificatori

ID = letter {letter | digit | “\_”}.

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

digit = “0”| “1”| “2”| “3”| “4”| “5”| “6”| “7”| “8”| “9”.

1. Constante

CONST\_STRING = “”” { digit | letter | special\_char}”””.

special\_char =“~”|”!”|”@”|”#”|”$”|”%”|”^”|”&”|” “ |”\*“|”(“|”)”|”\_”|”-”|”+”|”=”|“`”|”[”|”{“|”]”|”}”|”\”|”|”|”;”|”:”|”’”|””””|”,”|”<”|”.”|”>”|”?”|”/”.

CONST\_POSITIVE = digit\_positive { digit\_positive | ”0”} [ ”.” {digit\_positive | ”0”} digit\_positive ]

CONST = „0” | {(”+”|”-”) digit\_positive {digit\_positive | ”0”} [”.” { digit\_positive | ”0”} digit\_positive ]}.

Digit\_positive = “1”| “2”| “3”| “4”| “5”| “6”| “7”| “8”| “9”.

1. Key Words

“int”, “double”, “void”, “main”, “cout”, “cin”, “while”, “if”, “else”, “M\_PI”, “endl”.

1. Operators

“+”, “-“, “\*”, “<<”, ”>>”, “=”, “!=”, “>”, “<”, “<=”, “>=”, “==”, “[“, ”]”

1. Separators

“(“, “)”, “{“, “}”, “,”, “;”, “ “

EBNF :

Program : “int” “main” “(“ “)” “{“ {list\_decl} [list\_instr] “}”.

List\_decl= decl “;” {decl “;”}.

Decl = type list\_def.

Type = (“int” | “double”).

List\_def = def {“,” def}.

Def = ID [ “=” CONST ] | ID [ “[“ CONST\_POSITIVE“]” ].

List\_instr = instr “;” { instr “;”}.

Instr = assign | instr\_if | instr\_loop | read | write.

Assign = ID [“[“ CONST\_POSITIVE | Var “]” ] “=” var { operator\_arithm var }.

Var = ID | CONST | ID [ “[“ CONST\_POSITIVE “]” ].

Operator\_arithm = “+” | “-” | “\*”.

Instr\_if = “if” “(“ condition “)” “{“ list\_instr “}” [“else” “{“ list\_instr “}”.

Condition = var operator\_rel var.

Operator\_rel = “!=”, “>”, “<”, “<=”, “>=”, “==”.

Inst\_loop = “while” “(“ condition “)” “{“ list\_instr “}”.

Read = “cin” “>>” ID.

Write = “cout” writing {writing}.

Writing = “<<” ( var | “endl” | CONST\_STRING ).

Errors :

Int eroare1 ()

{

Int a, \_\_\_3;

Cin >> a

Cout << a;

}

Int eroare2 ()

{

Char litera =’a’;

Int b, c;

Cin >> b >> c;

}