1)

C++

<ID> -> <string>

<string>:<litera><string>|<litera>

<litera>:a|b|…|z

<antetBiblioteca>-># include <iostream>

<program>-><antetBiblioteca>”\n using namespace std;”<lista\_functii>

<lista\_functii>-><functie><lista\_functie>|<functie>|empty>

<functie> -> <antet> <lista\_instructiuni>

<antet> -> void|<tip simplu>|<Tip definit> ID(<lista\_parametri>)

<lista\_parametri> -><parametru>, <lista\_parametri>

<lista\_parametri> -> <empty>

<parametru>-><Tip> ID

<operatie> -> ID:<Tip><operator> ID:<Tip> | ID:<Tip><operator> <Tip> | <Tip><operator>

<operator>: <|>|==|<=|>=|%|\*|+|-|/

ID:<Tip> | <Tip><operator> <Tip>

<empty> ->

<parametru> ->  ID:<tip>

<lista\_instructiuni> -> <instr><lista\_instructiuni>

<instr> -> <selectie> | <ciclare> | <instr\_return>|<atribuire>|<consitie>

<instr\_return> -> return ID | CONST;

<declarare> : <tip> {("," ID)} ";"

<Tip>:<Tip simplu>|<Tip compus>

<Tip simplu>: int|float

<int>:<cifra\_nenula><int>|<cifra>

<cifra>:0|1|…|9

<cifra\_nenula>:1|2|…|9

<float>:<cifra><int>|<float>|<cifra><int>”.”<int>|<cifra><int>”.”<leading\_zeroes><int>

<leading\_zeroes>:0|0<leading\_zeroes>

<Tip compus>: struct ID{<lista declarari>};

<lista declarari>: <declarare>|<declarare><lista declarari>

<Atribuire>: ID:<Tip>=ID:<Tip>;|ID:<Tip>=CONST;|ID:<Tip>=<operatie>;

<Intrare>: cin>> ID:<Tip;

<Iesire>:cout<<ID <Tip><<’\n’;| cout<< ID:<Tip>;|cout<< <string>;|cout<< <apel\_functie><<’\n’;

<apel de functie>:<ID>()

<Selectie>:if(<conditie>){<lista instructiuni>}

<Ciclare>:while(conditie){<lista instrictiuni>}

<conditie>: ID<Tip><operator>ID:<Tip>

<program>   #include<iostream>using namespace std;<lista\_functii>

<lista\_functii>-><functie>;<lista\_functii>

<lista\_functii>-><functie>;

<lista\_instr>  <instr> ; <lista\_instr>

<lista\_instr>  <instr>

<instr>  <atribuire>

<atribuire>ID := <expr>

<expr>  <expr> + <termen>

<expr>  <termen>

<termen>  ID | CONST

<declaratie>  var <lista\_variabile> : integer;

<lista\_variabile>  ID | ID, <lista\_variabile>

<ID> ::= <string>

<string>::=<litera><string>|<litera>

<litera>::=a|b|…|z

<antetBiblioteca>::=# include <iostream>

<program>::=<antetBiblioteca>”\n using namespace std;”<lista\_functii>

<lista\_functii>::=<functie><lista\_functie>|<functie>|empty>

<functie> ::= <antet> <lista\_instructiuni>

<antet> ::= void|<tip simplu>|<Tip definit> ID(<lista\_parametri>)

<lista\_parametri> ::=<parametru>, <lista\_parametri>

<lista\_parametri> ::= <empty>

<parametru><Tip> ID

<operatie> ::= ID:<Tip><operator> ID:<Tip> | ID:<Tip><operator> <Tip> | <Tip><operator>

<operator>::= <|>|==|<=|>=|%|\*|+|-|/

ID:<Tip> | <Tip><operator> <Tip>

<empty> ::=

<parametru> ::=  ID:<tip>

<lista\_instructiuni> ::= <instr><lista\_instructiuni>

<instr> ::= <selectie> | <ciclare> | <instr\_return>|<atribuire>|<consitie>

<instr\_return> ::= return ID | CONST;

<declarare> ::= <tip> {("," ID)} ";"

<Tip>::=<Tip simplu>|<Tip compus>

<Tip simplu>::= int|float

<int>::=<cifra\_nenula><int>|<cifra>

<cifra>::=0|1|…|9

<cifra\_nenula>::=1|2|…|9

<float>::=<cifra><int>|<float>|<cifra><int>”.”<int>|<cifra><int>”.”<leading\_zeroes><int>

<leading\_zeroes>::=0|0<leading\_zeroes>

<Tip compus>: :=struct ID{<lista declarari>};

<lista declarari>::= <declarare>|<declarare><lista declarari>

<Atribuire>: :=ID:<Tip>=ID:<Tip>;|ID:<Tip>=CONST;|ID:<Tip>=<operatie>;

<Intrare>: :=cin>> ID:<Tip;

<Iesire>::=cout<<ID <Tip><<’\n’;| cout<< ID:<Tip>;|cout<< <string>;|cout<< <apel\_functie><<’\n’;

<apel de functie>:<ID>()

<Selectie>::=if(<conditie>){<lista instructiuni>}

<Ciclare>::=while(conditie){<lista instrictiuni>}

<conditie>::= ID<Tip><operator>ID:<Tip>