```
tp1_lemos.CErrorException: Undeclared variable a at line 10
at tp1_lemos.Visitor.checkVarName(Visitor.java:110)
at tp1_lemos.Visitor.enterTo_value(Visitor.java:84)
at tp1_lemos.CParser$To_valueContext.enterRule(CParser.java:1315)
at org.antlr.v4.runtime.tree.ParseTreeWalker.enterRule(ParseTreeWalker.java:66)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:49)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at org.antlr.v4.runtime.tree.ParseTreeWalker.walk(ParseTreeWalker.java:52)
at tp1_lemos.TP1_Lemos.main(TP1_Lemos.java:50)

int _x = 0;

double exec (double);

int main (void) {

    double y = 1, z = 3.5;

    for (_x = 5; _x <10; _x = _x + 1)

        y = exec(y) + exec(z);

    if (y <100) {

        y = y / 2;

        if (a == 0){}

    }

    else

        z = y + 5.5;

    return 0;

}

double exec (double asd){

    double exec (double valor) {

    int i = 10;

    while (i >= 5)

    valor -= _x - i;

    return valor;

    }

}
```

prog

stat — variable_declaration ; — int _x = to_value — digit — 0

stat — f.p ; — double exec (parametersList) — double

stat — f.d — int main (parametersList) — void — codeblock

{ stat — variable_declaration ; — double y = to_value , z = to_value — digit 1 — digit 3.5

for_loop — for (_x = 5 ; comparation ; assignation) stat
comparation — to_value < to_value — x — digit 10
assignation — _x assign_op to_value — = math_operation — _x + digit — 1
stat — assignation ; — y assign_op to_value — = math_operation — f_c + f_c
f_c — exec (inputParametersList) — to_value — y
f_c — exec (inputParametersList) — to_value — z

stat — if_condition — if (logic_op) codeblock else_condition
logic_op — comparation — to_value < to_value — digit 100 — y
codeblock — { stat stat }
stat — assignation ; — y assign_op to_value — = math_operation — y / digit — 2
stat — if_condition — if (logic_op) codeblock
logic_op — comparation — to_value == to_value — a — digit — 0
codeblock — { }
else_condition — else stat — assignation — z assign_op to_value — = math_operation — y + digit — 5.5

stat — ret ; — return to_value — digit — 0
}

stat — f.d — double exec (parametersList) codeblock
parametersList — double asd
codeblock — { stat }
f.d — double exec (parametersList) — double valor — codeblock

{ stat — variable_declaration ; — int i = to_value — digit — 10

stat — while_loop — while (logic_op)
logic_op — (comparation) — to_value >= to_value — i — digit — 5
stat — assignation — valor assign_op to_value — = math_operation — _x - 1

stat — ret ; — return to_value — valor
}