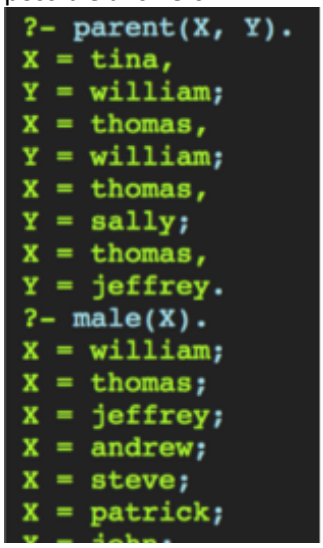
CodeQ and Prolog

The "CODE Q" application is an Intelligent tutoring system that enables "learning programming with automatically generated hints".

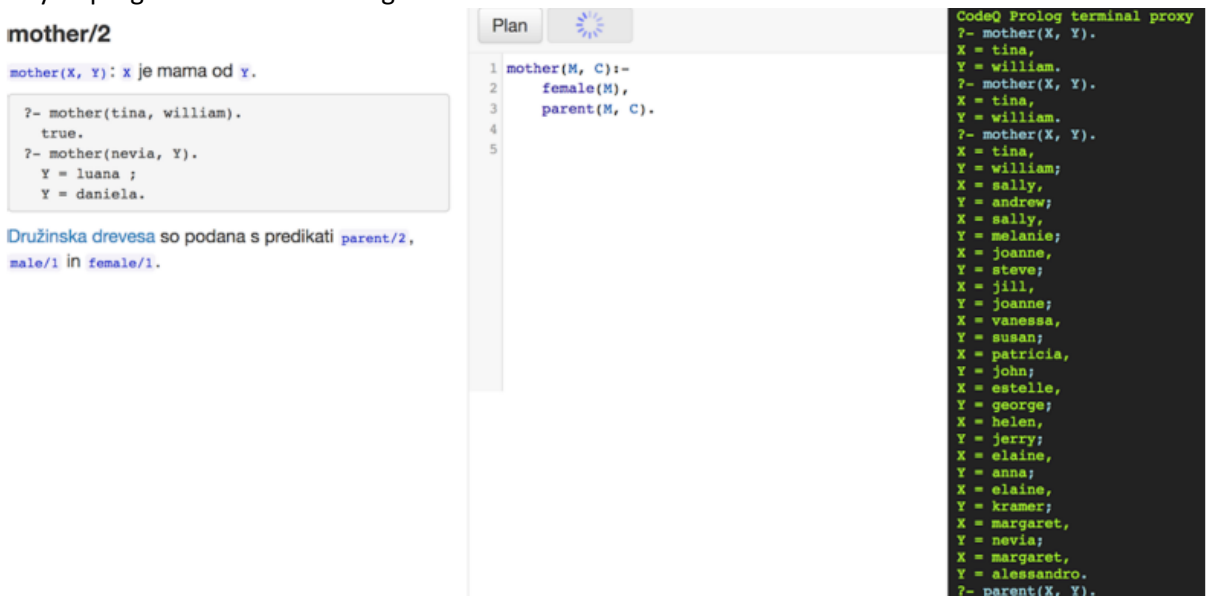
When using the CodeQ application, carefully read the instructions and follow them! I also recommend that you carefully read the "First Steps in Prolog" instructions that are given in the first task – writing the rules for different family relationships.

For each task, the CodeQ system contains prepared data and some defined relationships. So for example: "For tasks in this set, a family database is already defined. In Prolog, this information is represented by predicates *parent/2*, *male/1* and *female/1*."

First, test the already defined relationships by asking questions in Prolog, for example: "*parent(X, Y).*" or "*male(X).*" Repeatedly typing the semicolon ";" after responses are given by Prolog, you get all possible answers.



When you understand the problem well, continue with the task – by writing programs. The easiest way to progress is to follow the given instructions.



Writing Your/Additional Programs in Prolog

You can also use CodeQ to write and test your Prolog programs. In the text editor you program your program, while in the Console you ask Prolog questions. This picture shows an example of the program that we showed in the lecture.

Plan

```
1  %on(B1,B2).
2  on(a, b).
3  on(b, c).
4  on(c, table).
5  on(d, table).
6  on(e, table).
7
8  %see(B, X, Y).
9  see(a, 2, 5).
10 see(d, 5, 5).
11 see(e, 5, 2).
12
13 %above(B1, B2): B1 is above B2 in the same stack
14 above(B1, B2):-
15     on(B1, B2).
16
17 above(B1, B2):-
18     on(B1, B),
19     above(B, B2).
20
```

```
CodeQ Prolog terminal proxy
?- on(X, Y).
X = a,
Y = b.
?- see(X, Y, Z).
X = a,
Y = 2,
Z = 5;
X = d,
Y = Z = 5;
X = e,
Y = 5,
Z = 2.
?- above(a, X).
X = b;
X = c;
X = table;
false.
?- on(X, Y).
X = a,
Y = b;
X = b,
Y = c;
X = c,
Y = table;
X = d,
Y = table;
X = e,
Y = table.
?-
```