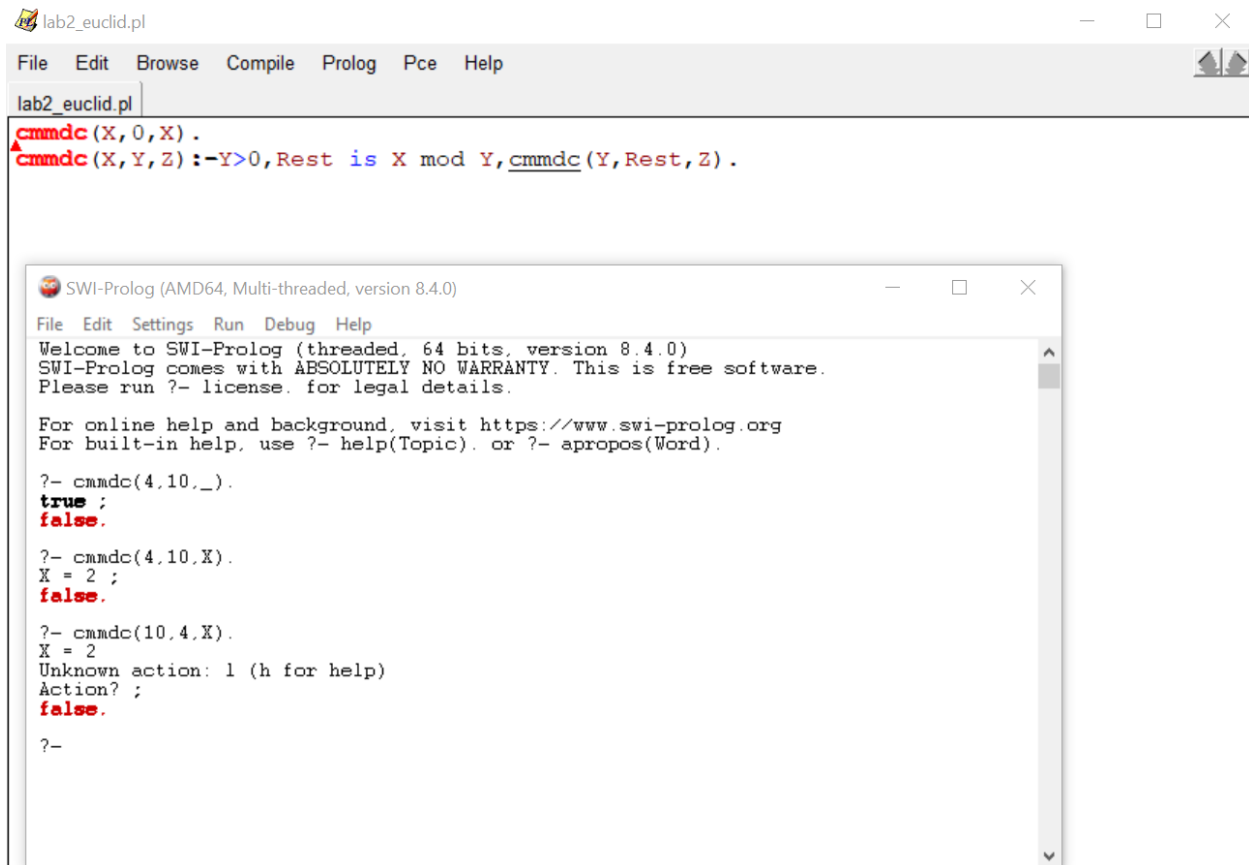


LABORATOR 2 IA - Recursivitate in prolog

Bumbescu Violeta-Andreea, AIA4.1_1A

1. Sa se scrie un predicat care calculeaza cel mai mare divizor comun pentru doua numere (folosind definitia recursiva a lui Euclid).



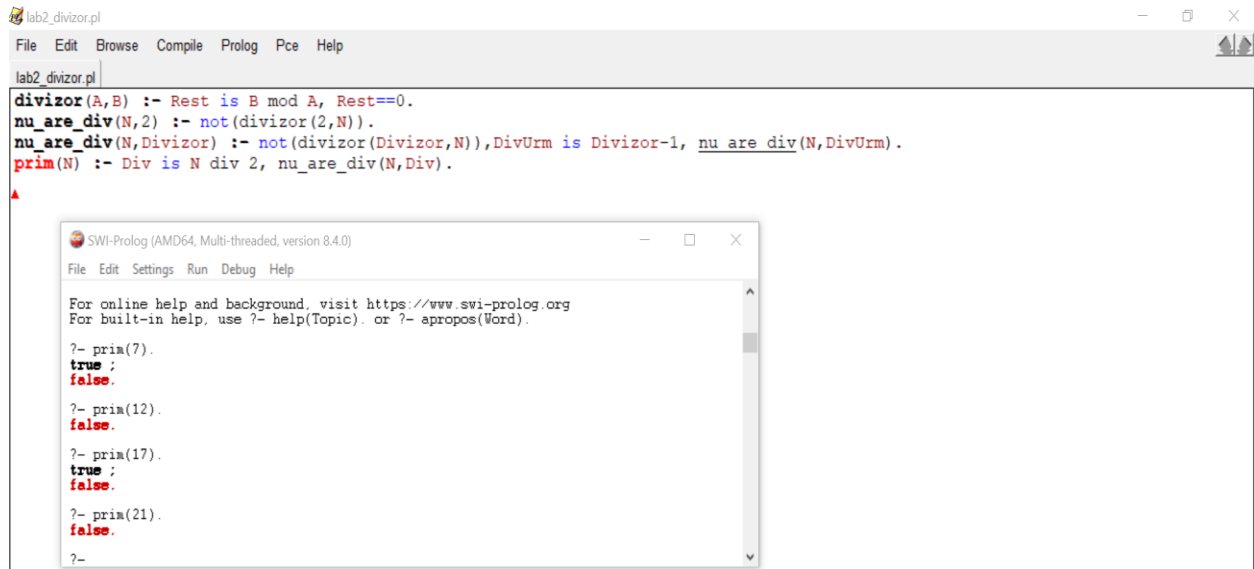
The screenshot shows a Prolog IDE window titled 'lab2_euclid.pl'. The code defines a predicate `cmmdc` for calculating the Greatest Common Divisor (GCD) using Euclid's algorithm. The code is as follows:

```
cmmdc(X, 0, X) .  
cmmdc(X, Y, Z) :- Y > 0, Rest is X mod Y, cmmdc(Y, Rest, Z) .
```

Below the code editor, a terminal window titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)' displays the execution results of several queries:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.  
Please run ?- license. for legal details.  
  
For online help and background, visit https://www.swi-prolog.org  
For built-in help, use ?- help(Topic). or ?- apropos(Word).  
  
?- cmmdc(4,10,_).  
true ;  
false.  
  
?- cmmdc(4,10,X).  
X = 2 ;  
false.  
  
?- cmmdc(10,4,X).  
X = 2  
Unknown action: 1 (h for help)  
Action? ;  
false.  
  
?-
```

2. Sa se scrie un predicat 'divizor' care testeaza daca un numar se imparte la altul. Pe baza acestui predicat sa se scrie un predicat 'prim' care spune daca un numar este prim.



```
lab2_divizor.pl
File Edit Browse Compile Prolog Pce Help
lab2_divizor.pl
divizor(A,B) :- Rest is B mod A, Rest==0.
nu_are_div(N,2) :- not(divizor(2,N)).
nu_are_div(N,Divizor) :- not(divizor(Divizor,N)), DivUrm is Divizor-1, nu_are_div(N,DivUrm).
prim(N) :- Div is N div 2, nu_are_div(N,Div).
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- prim(7).
true ;
false.

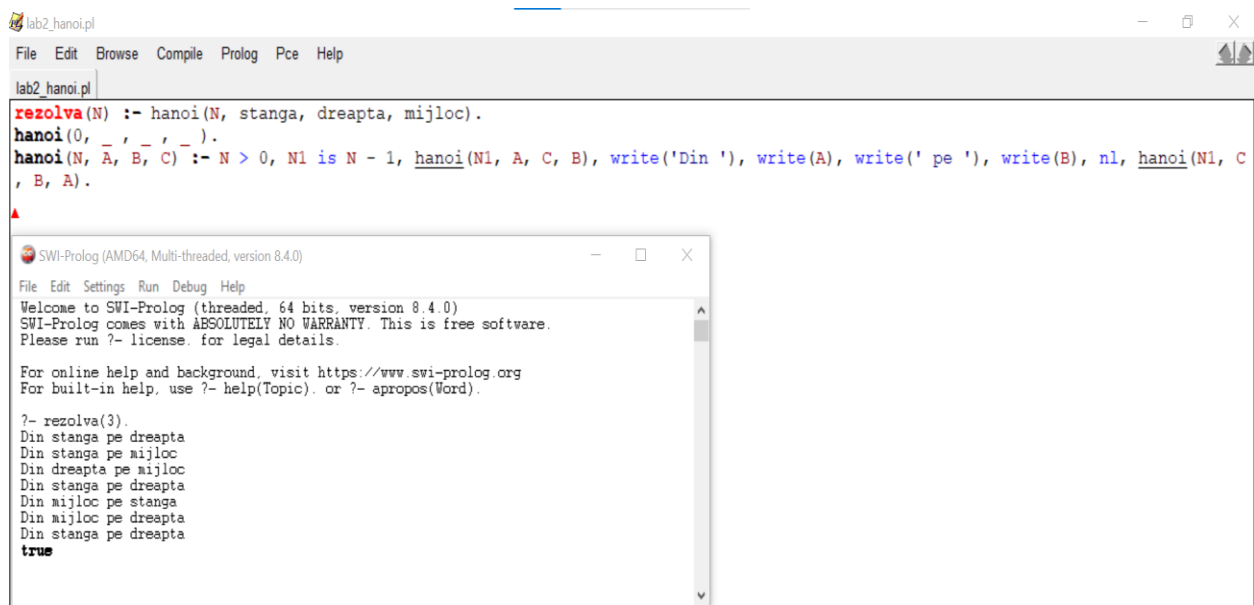
?- prim(12).
false.

?- prim(17).
true ;
false.

?- prim(21).
false.

?-
```

3. Sa se scrie un predicat rezolva_hanoi(N) care sa rezolve problema turnurilor din Hanoi cu N discuri.



```
lab2_hanoi.pl
File Edit Browse Compile Prolog Pce Help
lab2_hanoi.pl
rezolva(N) :- hanoi(N, stanga, dreapta, mijloc).
hanoi(0, _, _, _).
hanoi(N, A, B, C) :- N > 0, N1 is N - 1, hanoi(N1, A, C, B), write('Din '), write(A), write(' pe '), write(B), nl, hanoi(N1, C, B, A).
```

```
SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- rezolva(3).
Din stanga pe dreapta
Din stanga pe mijloc
Din dreapta pe mijloc
Din stanga pe dreapta
Din mijloc pe stanga
Din mijloc pe dreapta
Din stanga pe dreapta
true
```

1. Problema vecinilor

Ipoteze:

- 1) Ștefan este vecin cu Petre.
- 2) Ștefan este căsătorit cu o doamnă care lucrează la Spitalul de Urgență.
- 3) Petre este căsătorit cu o actriță care lucrează la Teatrul Național.
- 4) Ștefan este meloman și Petre este vânător.
- 5) Toți melomanii sunt sentimentali.
- 6) Toți vânătorii sunt mincinoși.
- 7) Actrițele iubesc bărbații sentimentali.
- 8) Soții au aceeași vecini.
- 9) Căsătoria și vecinătatea sunt relații simetrice.

Concluzie:

Îl iubește soția lui Petre pe Ștefan?

The screenshot shows a Prolog environment with a main editor window and a separate SWI-Prolog console window.

Main Editor Window:

- File Edit Browse Compile Prolog Pce Help
- lab2_problema-vecinilor.pl
- Code:

```
vecin(stefan, petre).
casatorit(stefan, sotie_stefan).
casatorit(petre, sotie_petre).
doctorita(sotie_stefan).
actrita(sotie_petre).
meloman(stefan).
vanator(petre).
sentimental(X):-meloman(X).
mincinos(X):-vanator(X).

vecin1(X,Y):-vecin(X,Y);vecin(Y,X).
vecinatate(X,Y):-vecin1(Z,Y),casatorit1(X,Z).

casatorit1(X,Y):-casatorit(X,Y);casatorit(Y,X).

iubeste(X,Y):-actrita(X),sentimental(Y),vecinatate(X,Y).
```

SWI-Prolog Console Window:

- SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)
- File Edit Settings Run Debug Help
- Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)
- SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software. Please run ?- license. for legal details.
- For online help and background, visit <https://www.swi-prolog.org>
- For built-in help, use ?- help(Topic). or ?- apropos(Word).
- ?- iubeste(sotie_petre,stefan).
- true.
- ?- iubeste(sotie_stefan,petre).
- false.
- ?-

Region consulted Line: 22

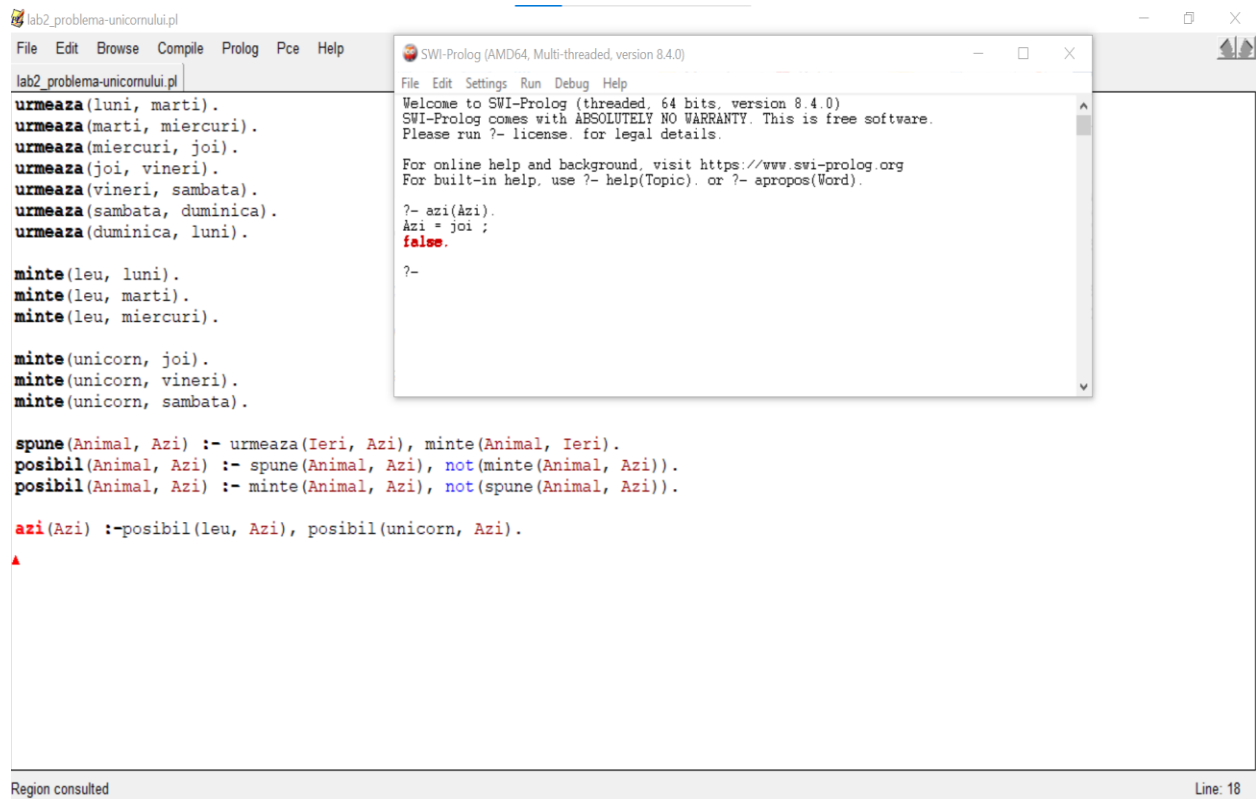
2. **Problema unicornului** este o problemă celebră formulată de Lewis Carroll în cartea sa "Alice în țara minunilor".

Ipoteze:

- 1) Leul minte luni, marți și miercuri și spune adevărul în toate celelalte zile.
- 2) Unicornul minte joi, vineri și sâmbătă și spune adevărul în toate celelalte zile.
- 3) Astăzi Leul spune: "Ieri a fost una din zilele în care eu mint."
- 4) Tot astăzi Unicornul spune: "Ieri a fost una din zilele în care eu mint."

Concluzie:

Ce zi este astăzi?



The screenshot shows a Prolog environment with two windows. The main window, titled 'lab2_problema-unicornului.pl', contains the following Prolog code:

```
urmeaza(luni, marti).
urmeaza(marti, miercuri).
urmeaza(miercuri, joi).
urmeaza(joi, vineri).
urmeaza(vineri, sambata).
urmeaza(sambata, duminica).
urmeaza(duminica, luni).

minte(leu, luni).
minte(leu, marti).
minte(leu, miercuri).

minte(unicorn, joi).
minte(unicorn, vineri).
minte(unicorn, sambata).

spune(Animal, Azi) :- urmeaza(Ieri, Azi), minte(Animal, Ieri).
posibil(Animal, Azi) :- spune(Animal, Azi), not(minte(Animal, Azi)).
posibil(Animal, Azi) :- minte(Animal, Azi), not(spune(Animal, Azi)).

azi(Azi) :-posibil(leu, Azi), posibil(unicorn, Azi).
```

The output window, titled 'SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)', shows the following output:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.0)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- azi(Azi).
Azi = joi ;
false.
?-
```

At the bottom of the main window, it says 'Region consulted' and 'Line: 18'.