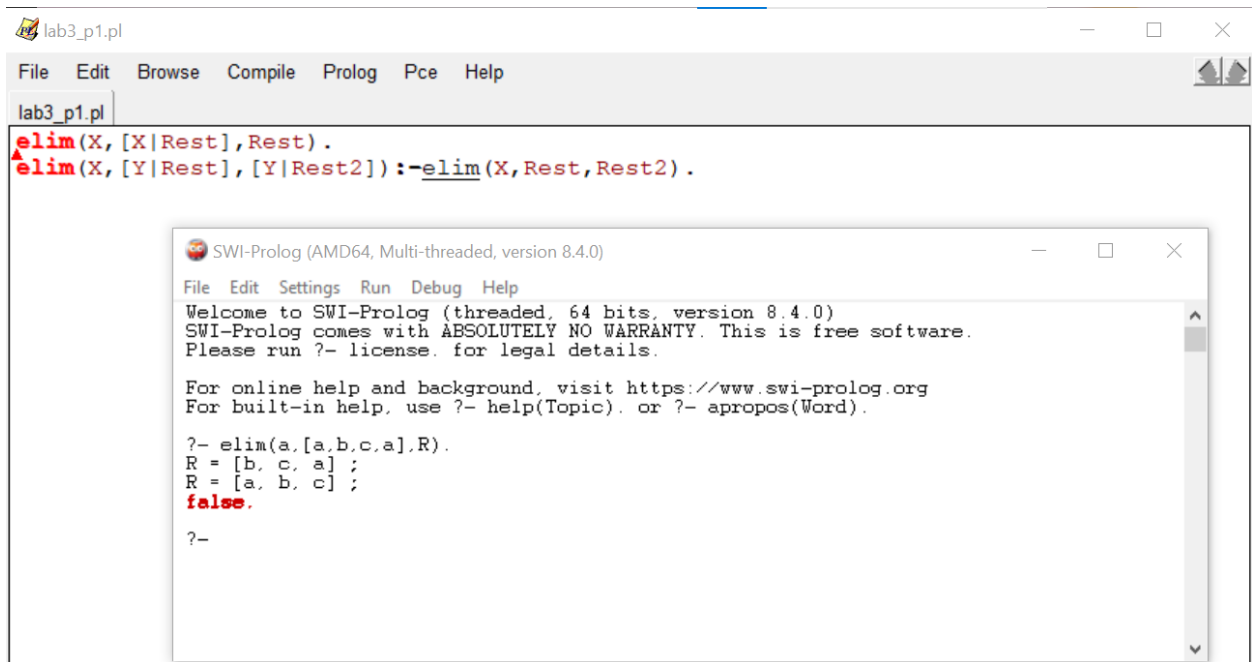


Laborator 3 IA - Prelucrarea listelor in Prolog

Bumbescu Violeta-Andreea, AIA 4.1.1_A

1. **Eliminarea unui obiect dintr-o lista.** Scrieti un predicat care elimina un obiect dintr-o lista. Astfel, `elim(a, [a, b, c], L)` va returna în L lista [b, c].



The screenshot shows a Prolog IDE window titled 'lab3_p1.pl'. The code defines two predicates: `elim(X, [X|Rest], Rest).` and `elim(X, [Y|Rest], [Y|Rest2]) :- elim(X, Rest, Rest2).`. Below the code, a SWI-Prolog console window is open, displaying the following text:

```
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).

?- elim(a,[a,b,c,a],R).
R = [b, c, a] ;
R = [a, b, c] ;
false.
?-
```

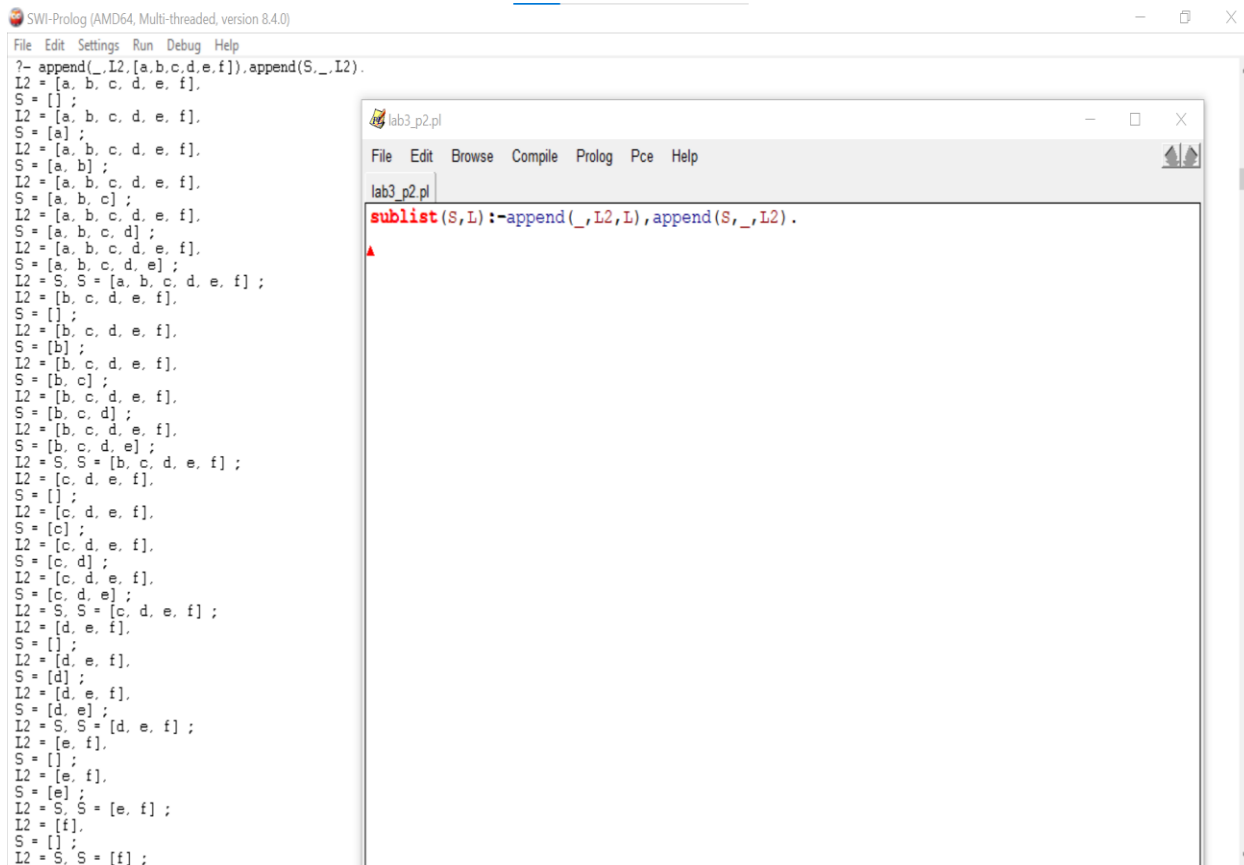
2. **Incluziunea listelor.** Scrieti un predicat care este adevarat daca o lista este sublistata alteia. De exemplu, `sublist([c, d, e], [a, b, c, d, e, f])` este adevarat, iar `sublist([b, c, e], [a, b, c, d, e, f])` este fals.

Indicatie: Puteti folosi predicatul `append`. O lista `S` este sublistata a listei `L` daca:

Exista o descompunere a lui `L` în `L1` si `L2`

si

Exista o descompunere a lui `L2` în `S` si `L3`.



The image shows a Prolog IDE window titled "SWI-Prolog (AMD64, Multi-threaded, version 8.4.0)". The main window displays a series of queries and their results, demonstrating the `sublist` predicate. The queries are as follows:

```
?- append(_,L2,[a,b,c,d,e,f]),append(S,_,L2).
L2 = [a,b,c,d,e,f],
S = [] ;
L2 = [a,b,c,d,e,f],
S = [a] ;
L2 = [a,b,c,d,e,f],
S = [a,b] ;
L2 = [a,b,c,d,e,f],
S = [a,b,c] ;
L2 = [a,b,c,d,e,f],
S = [a,b,c,d] ;
L2 = [a,b,c,d,e,f],
S = [a,b,c,d,e] ;
L2 = S, S = [a,b,c,d,e,f] ;
L2 = [b,c,d,e,f],
S = [] ;
L2 = [b,c,d,e,f],
S = [b] ;
L2 = [b,c,d,e,f],
S = [b,c] ;
L2 = [b,c,d,e,f],
S = [b,c,d] ;
L2 = [b,c,d,e,f],
S = [b,c,d,e] ;
L2 = S, S = [b,c,d,e,f] ;
L2 = [c,d,e,f],
S = [] ;
L2 = [c,d,e,f],
S = [c] ;
L2 = [c,d,e,f],
S = [c,d] ;
L2 = [c,d,e,f],
S = [c,d,e] ;
L2 = S, S = [c,d,e,f] ;
L2 = [d,e,f],
S = [] ;
L2 = [d,e,f],
S = [d] ;
L2 = [d,e,f],
S = [d,e] ;
L2 = S, S = [d,e,f] ;
L2 = [e,f],
S = [] ;
L2 = [e,f],
S = [e] ;
L2 = S, S = [e,f] ;
L2 = [f],
S = [] ;
L2 = S, S = [f] ;
```

A separate window titled "lab3_p2.pl" shows the definition of the `sublist` predicate:

```
sublist(S,L):-append(_,L2,L),append(S,_,L2).
```

3. **Liniazarea listelor.** Scrie predicatul `liniar(ListaListe, Lista)`, unde `ListeListe` este o lista de elemente care pot fi rândul lor liste, iar în `Lista` se construiește liniazarea listei `ListeListe`. Astfel, `liniar([1, 2, [3, 4], [5, [6, 7], [[8], 9]]], L)` va returna în `L` lista `[1, 2, 3, 4, 5, 6, 7, 8, 9]`.

```
lab3-p3.pl
File Edit Browse Compile Prolog Pce Help
lab3-p3.pl
liniazare([], []).
liniazare([_|Rest], Rez) :- liniazare(Rest, Rez), !.
liniazare([X|Rest], [X|Rez]) :- X \= [],
    X = [_|_],
    liniazare(Rest, Rez).
liniazare([X|Rest|Rest2], Rez) :- liniazare([X, Rest|Rest2], Rez).
```

```
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).

?- liniazare([a.b.c.d.e.f],[c.d.e]).
false.
?- liniazare([c.d.e],[a.b.c.d.e.f]).
false.
?-
```

4. Scrieti un predicat `descomp(N, Lista)` care primește un număr întreg `N` și întoarce o listă a factorilor primi ai numărului `N`; de exemplu: `descomp(12, [2, 2, 3])` este adevărat.

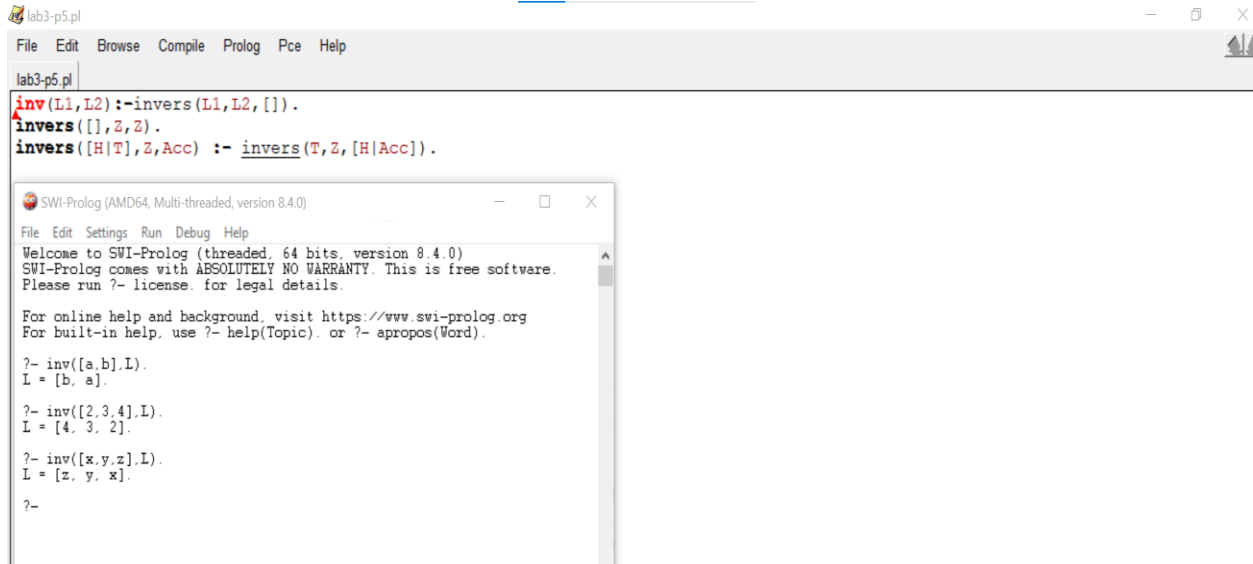
```
lab3_p4.pl
File Edit Browse Compile Prolog Pce Help
lab3_p4.pl
descomp(N, L) :- factprim(N, L, 2).
factprim(1, [], _).
factprim(N, [Divizor | Lista], Divizor) :- N > 1, 0 is N mod Divizor, N1 is N div Divizor, factprim(N1, Lista, Divizor).
factprim(N, Lista, Divizor) :- N > 1, not(0 is N mod Divizor), D1 is Divizor+1, factprim(N, Lista, D1).
```

```
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).

?- descomp(12,[2,2,3]).
true ;
false.
?- descomp(18,[9,2]).
false.
?-
```

5. Scrieti un predicat `invers(Lista, ListaInversata)` care inverseaza elementele unei liste; sa se scrie doua variante ale predicatului de inversare a unei liste: o varianta în care lista inversata este calculata pe ramura de revenire din recursivitate și o varianta în care lista inversata este calculata pe ramura de avans în recursivitate.



```
lab3-p5.pl
File Edit Browse Compile Prolog Pce Help
lab3-p5.pl
invers(L1,L2):-invers(L1,L2,[]).
invers([],Z,Z).
invers([H|T],Z,Acc):-invers(T,Z,[H|Acc]).
```

```
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).

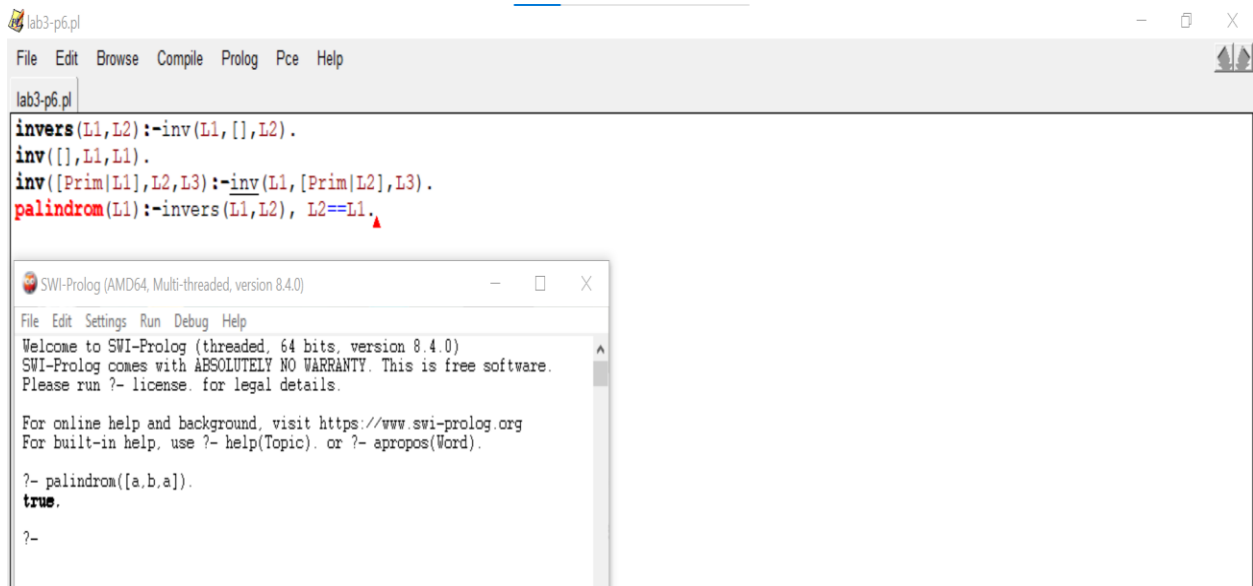
?- inv([a,b],L).
L = [b, a].

?- inv([2,3,4],L).
L = [4, 3, 2].

?- inv([x,y,z],L).
L = [z, y, x].

?-
```

6. Scrieti un predicat `palindrom(Lista)` care verifica daca o lista este palindrom. (Un palindrom este o secventa care, daca este parcursa de la stânga la dreapta sau de la dreapta la stânga, este identica; de exemplu: [a, b, c, b, a] sau [a, b, c, c, b, a].)



```
lab3-p6.pl
File Edit Browse Compile Prolog Pce Help
lab3-p6.pl
invers(L1,L2):-inv(L1,[],L2).
inv([],L1,L1).
inv([Prim|L1],L2,L3):-inv(L1,[Prim|L2],L3).
palindrom(L1):-invers(L1,L2), L2==L1.
```

```
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).

?- palindrom([a,b,a]).
true.

?-
```

