Laborator 2 – Prolog 1

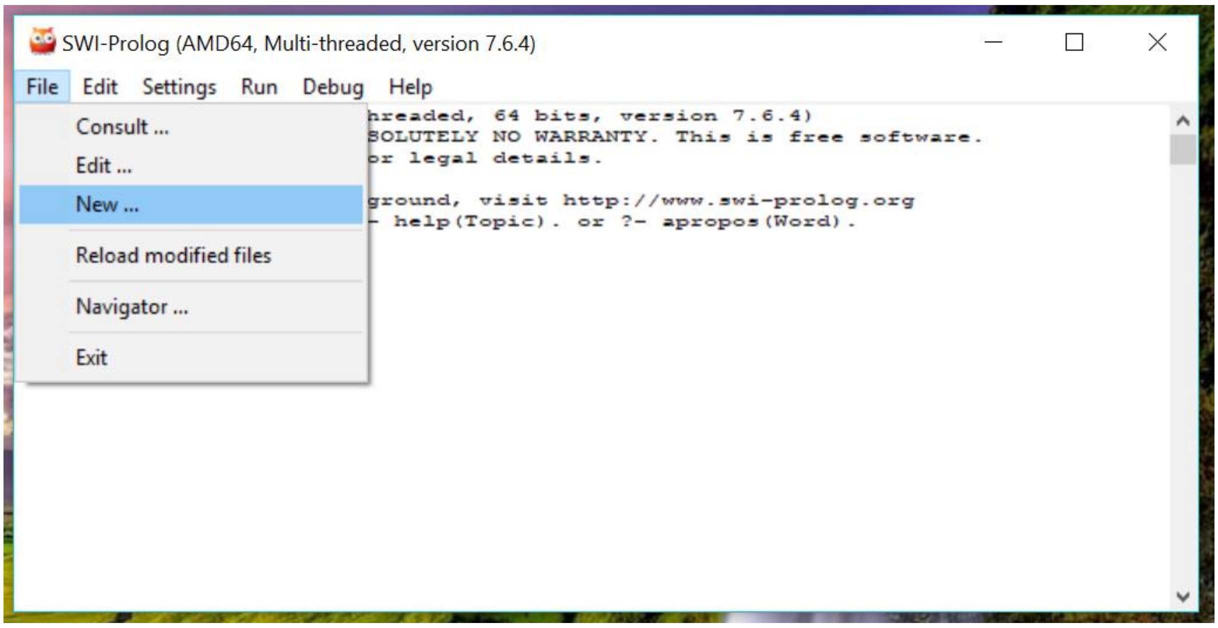
Lab 2 **P1**	Pozitie in grupa in lista studentilor	1 16	2 17	3 18	4 19	5 20	6 21	7 22	8 23	9 24	10 25	11 26	12 27	13 28	14 29	15 30
	Nr problema asignat	**12**	**14**	**15**	**1**	**2**	**4**	**6**	**8**	**1**	**11**	**9**	**7**	**5**	**3**	**8**
	Link enunturi probleme	http://www.cs.ubbcluj.ro/~hfpop/teaching/2018/pfl/lab/p1.pdf														
	Termen de predare	Sapt. 3 - 4														

Implementations will be done in:

- SWI Prolog (de instalat)
  or
- SWISH Prolog (online)

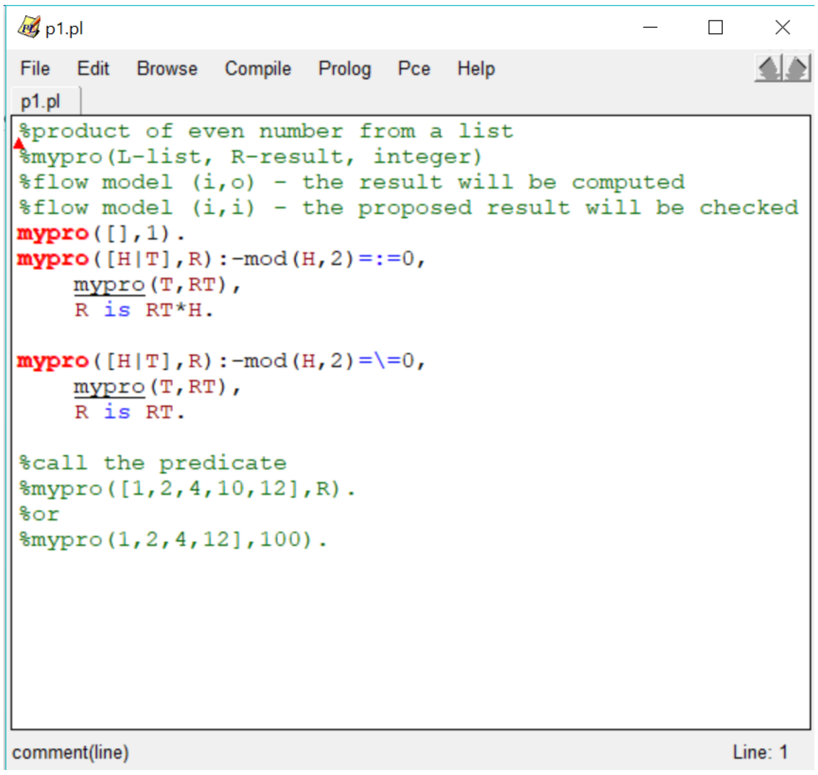
Problema 1 – model in **SWI PROLOG**

Download from here: http://www.swi-prolog.org/Download.html and install.



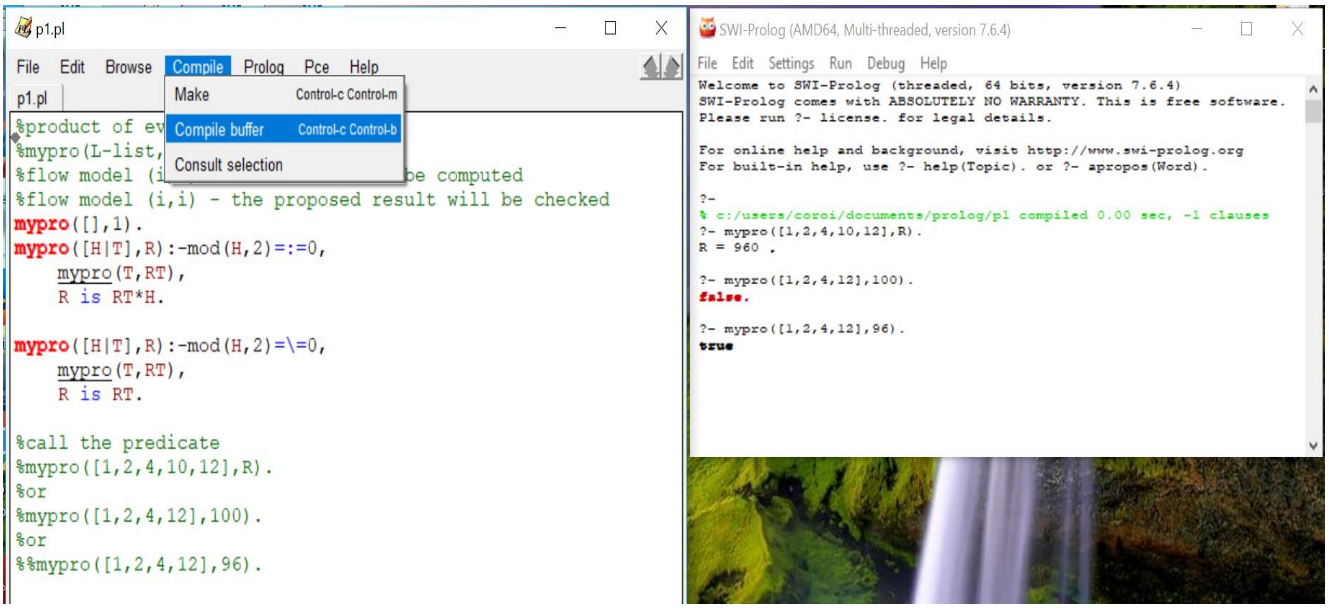
```
hreaded, 64 bits, version 7.6.4)
BOLUTELY NO WARRANTY. This is free software.
or legal details.

ground, visit http://www.swi-prolog.org
- help(Topic). or ?- apropos(Word).
```

Edit your code:



```
%product of even number from a list
%mypro(L-list, R-result, integer)
%flow model (i,o) - the result will be computed
%flow model (i,i) - the proposed result will be checked
mypro([],1).
mypro([H|T],R):-mod(H,2)=:=0,
    mypro(T,RT),
    R is RT*H.

mypro([H|T],R):-mod(H,2)=\=0,
    mypro(T,RT),
    R is RT.

%call the predicate
%mypro([1,2,4,10,12],R).
%or
%mypro(1,2,4,12],100).
```

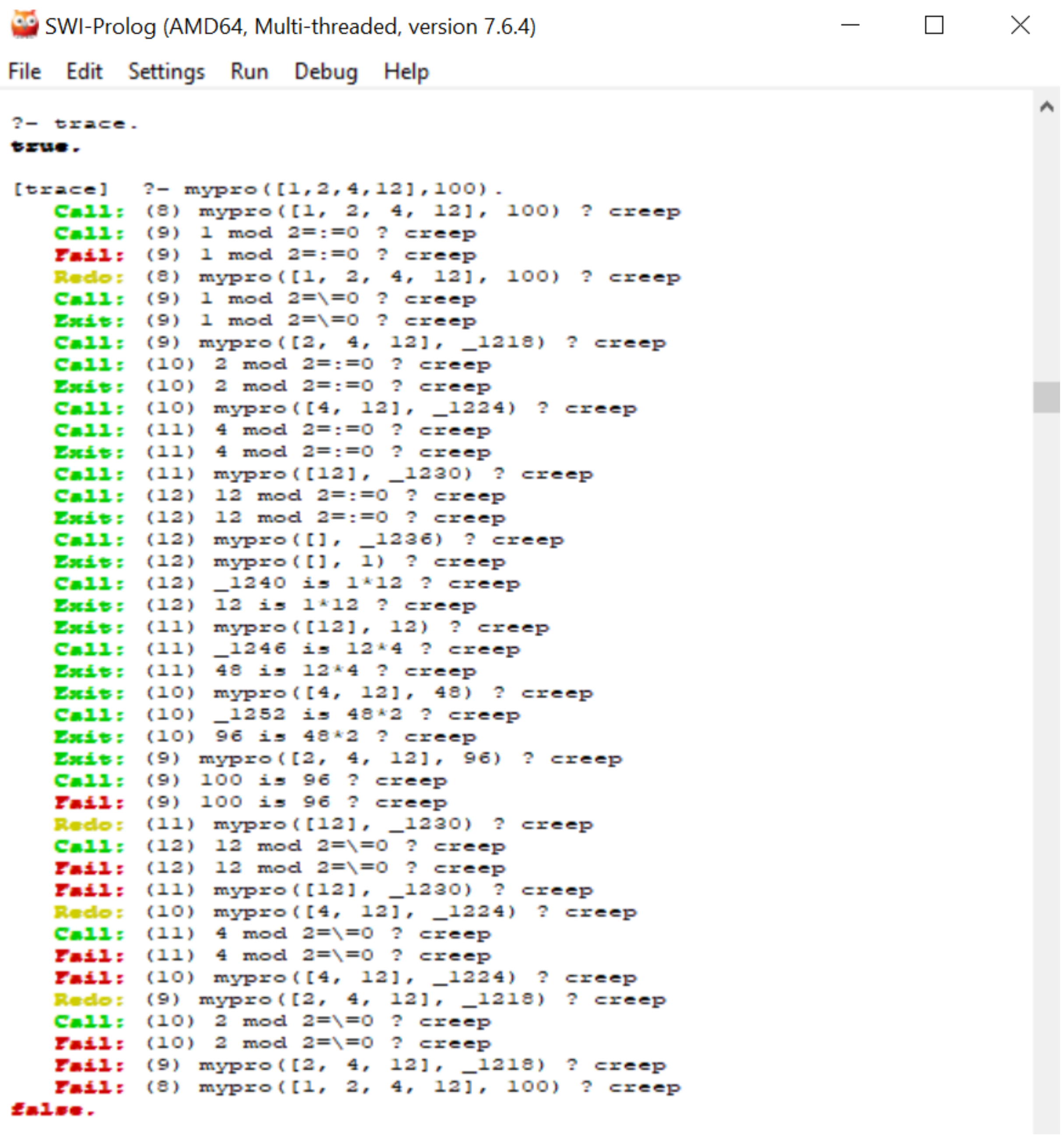
Compile (1) and run (2)


p1.pl
File Edit Browse Compile Prolog Pce Help
p1.pl
Make Control-c Control-m
Compile buffer Control-c Control-b
Consult selection
SWI-Prolog (AMD64, Multi-threaded, version 7.6.4)
File Edit Settings Run Debug Help

```
%product of ev
%mypro(L-list,
%flow model (i                          be computed
%flow model (i,i) - the proposed result will be checked
mypro([],1).
mypro([H|T],R):-mod(H,2)=:=0,
    mypro(T,RT),
    R is RT*H.

mypro([H|T],R):-mod(H,2)=\=0,
    mypro(T,RT),
    R is RT.

%call the predicate
%mypro([1,2,4,10,12],R).
%or
%mypro([1,2,4,12],100).
%or
%%mypro([1,2,4,12],96).
```

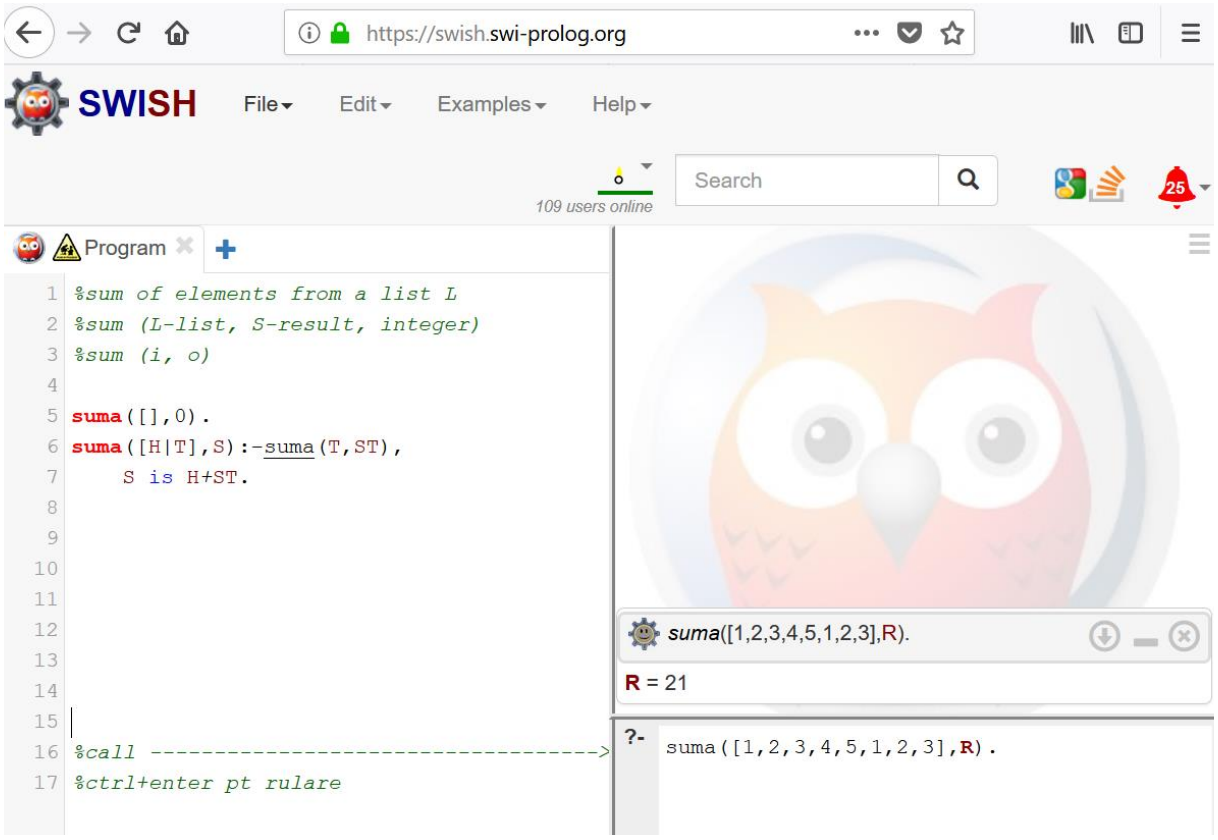
```
Welcome to SWI-Prolog (threaded, 64 bits, version 7.6.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

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

?-
% c:/users/coroi/documents/prolog/pl compiled 0.00 sec, -1 clauses
?- mypro([1,2,4,10,12],R).
R = 960 .

?- mypro([1,2,4,12],100).
false.

?- mypro([1,2,4,12],96).
true
```

Trace for activate step by step execution:


SWI-Prolog (AMD64, Multi-threaded, version 7.6.4)
File Edit Settings Run Debug Help

```
?- trace.
true.

[trace]  ?- mypro([1,2,4,12],100).
   Call: (8) mypro([1, 2, 4, 12], 100) ? creep
   Call: (9) 1 mod 2=:=0 ? creep
   Fail: (9) 1 mod 2=:=0 ? creep
   Redo: (8) mypro([1, 2, 4, 12], 100) ? creep
   Call: (9) 1 mod 2=\=0 ? creep
   Exit: (9) 1 mod 2=\=0 ? creep
   Call: (9) mypro([2, 4, 12], _1218) ? creep
   Call: (10) 2 mod 2=:=0 ? creep
   Exit: (10) 2 mod 2=:=0 ? creep
   Call: (10) mypro([4, 12], _1224) ? creep
   Call: (11) 4 mod 2=:=0 ? creep
   Exit: (11) 4 mod 2=:=0 ? creep
   Call: (11) mypro([12], _1230) ? creep
   Call: (12) 12 mod 2=:=0 ? creep
   Exit: (12) 12 mod 2=:=0 ? creep
   Call: (12) mypro([], _1236) ? creep
   Exit: (12) mypro([], 1) ? creep
   Call: (12) _1240 is 1*12 ? creep
   Exit: (12) 12 is 1*12 ? creep
   Exit: (11) mypro([12], 12) ? creep
   Call: (11) _1246 is 12*4 ? creep
   Exit: (11) 48 is 12*4 ? creep
   Exit: (10) mypro([4, 12], 48) ? creep
   Call: (10) _1252 is 48*2 ? creep
   Exit: (10) 96 is 48*2 ? creep
   Exit: (9) mypro([2, 4, 12], 96) ? creep
   Call: (9) 100 is 96 ? creep
   Fail: (9) 100 is 96 ? creep
   Redo: (11) mypro([12], _1230) ? creep
   Call: (12) 12 mod 2=\=0 ? creep
   Fail: (12) 12 mod 2=\=0 ? creep
   Fail: (11) mypro([12], _1230) ? creep
   Redo: (10) mypro([4, 12], _1224) ? creep
   Call: (11) 4 mod 2=\=0 ? creep
   Fail: (11) 4 mod 2=\=0 ? creep
   Fail: (10) mypro([4, 12], _1224) ? creep
   Redo: (9) mypro([2, 4, 12], _1218) ? creep
   Call: (10) 2 mod 2=\=0 ? creep
   Fail: (10) 2 mod 2=\=0 ? creep
   Fail: (9) mypro([2, 4, 12], _1218) ? creep
   Fail: (8) mypro([1, 2, 4, 12], 100) ? creep
false.
```

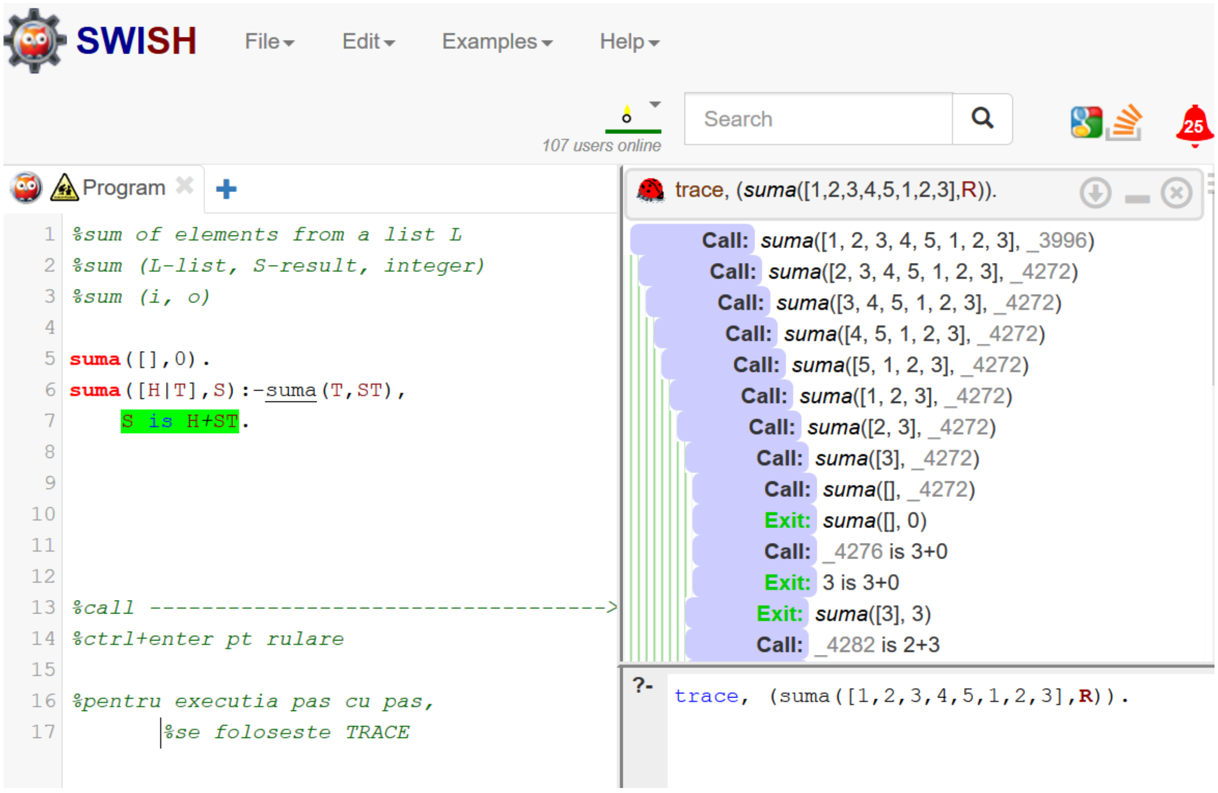
Problema 2 – model SWISH Prolog



```
%sum of elements from a list L
%sum (L-list, S-result, integer)
%sum (i, o)

suma([],0).
suma([H|T],S):-suma(T,ST),
    S is H+ST.








%call ------------------------------------>
%ctrl+enter pt rulare
```

```
suma([1,2,3,4,5,1,2,3],R).
R = 21
?- suma([1,2,3,4,5,1,2,3],R).
```

Step by step execution – using **TRACE,** here in Swish Prolog

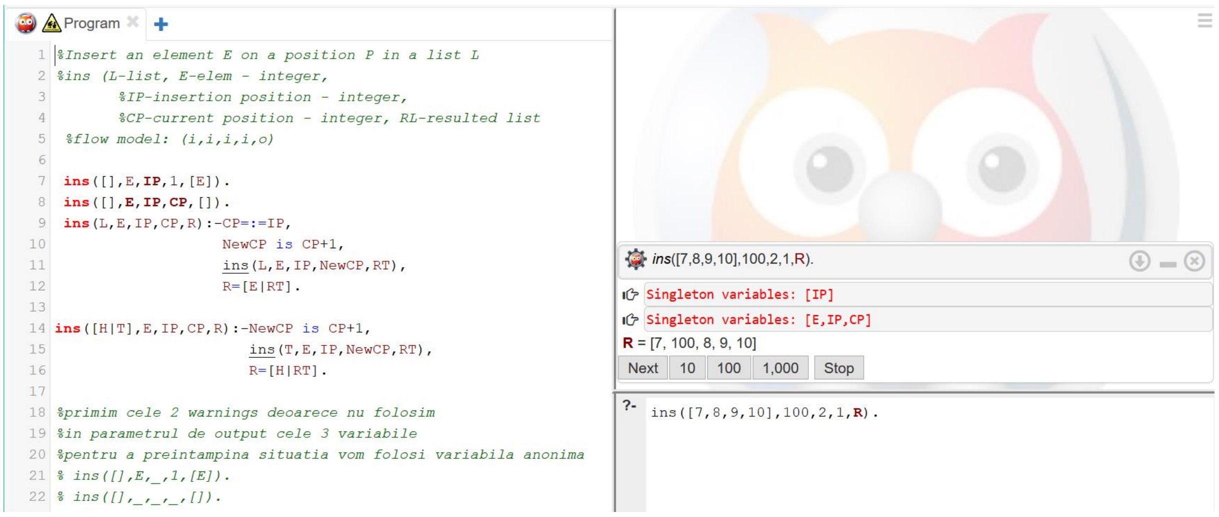


```
%sum of elements from a list L
%sum (L-list, S-result, integer)
%sum (i, o)

suma([],0).
suma([H|T],S):-suma(T,ST),
    S is H+ST.





%call ------------------------------------>
%ctrl+enter pt rulare

%pentru executia pas cu pas,
        %se foloseste TRACE
```

```
trace, (suma([1,2,3,4,5,1,2,3],R)).
 Call: suma([1, 2, 3, 4, 5, 1, 2, 3], _3996)
  Call: suma([2, 3, 4, 5, 1, 2, 3], _4272)
   Call: suma([3, 4, 5, 1, 2, 3], _4272)
    Call: suma([4, 5, 1, 2, 3], _4272)
     Call: suma([5, 1, 2, 3], _4272)
      Call: suma([1, 2, 3], _4272)
       Call: suma([2, 3], _4272)
        Call: suma([3], _4272)
         Call: suma([], _4272)
         Exit: suma([], 0)
         Call: _4276 is 3+0
         Exit: 3 is 3+0
        Exit: suma([3], 3)
        Call: _4282 is 2+3
?- trace, (suma([1,2,3,4,5,1,2,3],R)).
```

Problema 3 – model in SWISH or in SWI PROLOG, is up to you :)

(aici folosim nume de variabile)

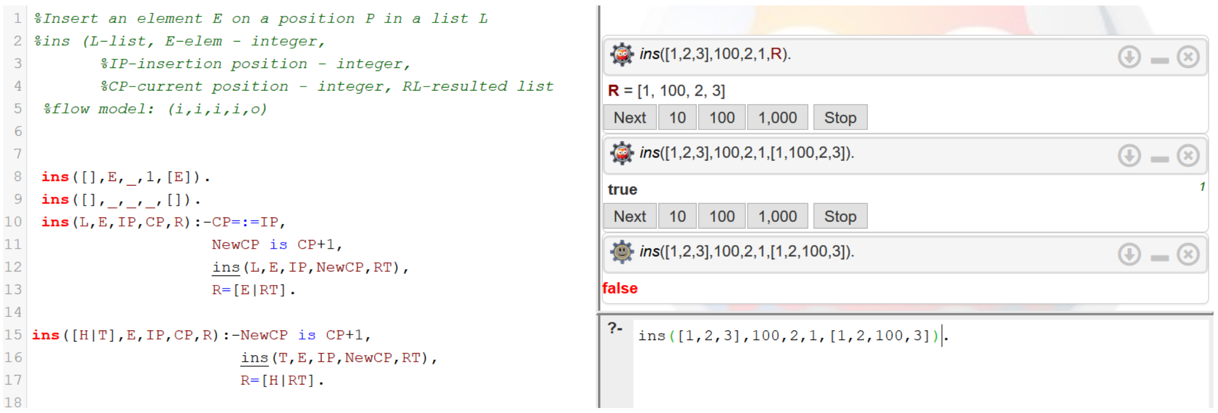


```
%Insert an element E on a position P in a list L
%ins (L-list, E-elem - integer,
       %IP-insertion position - integer,
       %CP-current position - integer, RL-resulted list
 %flow model: (i,i,i,i,o)

 ins([],E,IP,1,[E]).
 ins([],E,IP,CP,[]).
 ins(L,E,IP,CP,R):-CP=:=IP,
                   NewCP is CP+1,
                   ins(L,E,IP,NewCP,RT),
                   R=[E|RT].

ins([H|T],E,IP,CP,R):-NewCP is CP+1,
                      ins(T,E,IP,NewCP,RT),
                      R=[H|RT].

%primim cele 2 warnings deoarece nu folosim
%in parametrul de output cele 3 variabile
%pentru a preintampina situatia vom folosi variabila anonima
% ins([],E,_,1,[E]).
% ins([],_,_,_,[]).
```

```
ins([7,8,9,10],100,2,1,R).
Singleton variables: [IP]
Singleton variables: [E,IP,CP]
R = [7, 100, 8, 9, 10]

?- ins([7,8,9,10],100,2,1,R).
```

Aici folosim variabila anonima:

Si verificam functionalitatea predicatului si folosind modelul de flow (i, i, i, i, i) – iar ca rezultat vom avea True/False



```
%Insert an element E on a position P in a list L
%ins (L-list, E-elem - integer,
       %IP-insertion position - integer,
       %CP-current position - integer, RL-resulted list
 %flow model: (i,i,i,i,o)


 ins([],E,_,1,[E]).
 ins([],_,_,_,[]).
 ins(L,E,IP,CP,R):-CP=:=IP,
                   NewCP is CP+1,
                   ins(L,E,IP,NewCP,RT),
                   R=[E|RT].

ins([H|T],E,IP,CP,R):-NewCP is CP+1,
                      ins(T,E,IP,NewCP,RT),
                      R=[H|RT].

```

```
ins([1,2,3],100,2,1,R).
R = [1, 100, 2, 3]

ins([1,2,3],100,2,1,[1,100,2,3]).
true                                          1

ins([1,2,3],100,2,1,[1,2,100,3]).
false

?- ins([1,2,3],100,2,1,[1,2,100,3]).
```

Va rog sa va instalati si sa verificati functionalitatea predicatelor de mai sus, urmarind cel putin pentru o problema executia pas cu pas.