

LISTE ÎN PROLOG

I. Definirea listelor în PROLOG

- a. Lista vidă: []
- b. Lista nevidă cu valori întregi: [10,20,30,40,50]
- c. În Prolog – definire recursivă de forma: [H|T]
[H|T] semnifică:
H- primul element din listă (Head), iar T- restul listei (Tail)
Exemplu: [1,2,3,4,5] => H=1, iar T=[2,3,4,5]
- d. Declararea domeniului listă în PROLOG:

```
domains
    lista=tip_date*
```

Exemplu: lista= integer*, sau matrice=lista*

II. Prelucrări asupra listelor

1. Verificarea apartenenței unui element într-o listă de numere reale.
(X in L, unde L=[H1, H2, H3, ..., Hn])

Program PROLOG

```
domains
    lista=real*
predicates
    member(real, lista)
clauses
    member(X, [X|_]):-!. /*member(X, [X|_]):-!. */
    member(X, [_|T]):- member(X,T). /*member(X, [_|T]):- member(X,T). */
```

GOAL: member(7, [3, 7, -2, 5, 9, 0]) member(-7, [3, 7, -2, 5, 9, 0])

2. Lungimea (dimensiunea) unei liste date.

Program PROLOG

```
domains
    lista=real*
predicates
    lungime(lista, integer)
```

clauses

lungime([], 0).

lungime([H|T], L):-lungime(T, L1), L=L1+1.

GOAL: lungime([3, 7, -2, 5, 9, 0], Lungime)

3. Suma elementelor dintr-o listă de numere întregi.

Program PROLOG

domains

lista=integer*

predicates

suma(lista, real)

clauses

suma([], 0).

suma([H|T], S):-suma(T, S1), S=S1+H.

GOAL: suma([3, 7, -2, 5, 9, 0], Suma)

4. Concatenarea (alipirea) a două liste de numere reale (LR=L1L2).

L1=[H1,H2,...,Hn], L2=[Z1,Z2,...,Zm], LR=[H1,H2,...,Hn,Z1,Z2,...,Zm] cu
precizarea LR=[HR|TR], unde HR=H1, iar TR=[T1,L2], T1=restul listei L1.
(suplimentar este dezvoltat predicatul de concatenare a a 3 liste)

domains

lista=integer*

predicates

concatenare(lista, lista, lista)

concatenare3(lista,lista,lista,lista)

clauses

concatenare([], L2, L2).

concatenare([H1|T1], L2, [H1|TR]):-concatenare(T1, L2, TR).

concatenare3(L1, L2, L3, R):-

concatenare(L1,L2,R1),concatenare(R1,L3,R).

GOAL: concatenare([3,7,-2,5,9,0], [3,4,5,6,7], Rez)

Concatenare3([3,7,-2,5,9,0], [3,4,5,6,7], [3,2,1,0,6], R)

5. Numărul de apariții al unui element într-o listă de numere întregi.

Program PROLOG

domains

```

lista=integer*
predicates
    count(integer, lista, integer)
clauses
    count(X, [], 0).
    count(X, [X|T], R):- count(X, T, R1), R=R1+1, !.
    count(X, [_|T], R):- count(X, T, R).

```

GOAL: count(7, [3,7,-2,7,7,0,7,7], Rez)

6. Ștergerea unui element dintr-o listă de numere reale.

a. ștergerea primei apariții în lista dată

```

sterge1(X, [X|T], T):-!.
sterge1(X, [_|T], [_|T1]):- sterge1(X, T, T1).

```

b. ștergerea tuturor aparițiilor în lista dată

```

sterge2(X, [], []).
sterge2(X, [X|T], R):- sterge2(X,T,R), !.
sterge2(X, [_|T], [_|TR]):- sterge2(X, T, TR).

```

GOAL: sterge1(7, [3,7,-2,7,7,0,7,7], Rez) | sterge2(7, [3,7,-2,7,7,0,7,7], Rez)

7. Împărțirea unei liste de numere reale în două liste, o listă cu valori pozitive, iar cealaltă cu valori negative.

Program PROLOG

```

domains
    lista=real*
predicates
    impartire(lista, lista, lista)
clauses
    impartire([], [], []).
    impartire([H|T], [H|TPoz], LNeg):- H>=0, impartire(T, TPoz, LNeg), !.
    impartire([H|T], LPoz, [H|TNeg]):-impartire(T, LPoz, TNeg).

```

GOAL: impartire([3,7,-2,7,-7,0,7,-8], LPoz, LNeg)

III. Aplicații folosind listele

1. Împărțirea unei liste de numere reale în două liste în funcție de un parametru real constant k.