

Laborator02

Petculescu Mihai-Silviu

Laborator02

Petculescu Mihai-Silviu

Aplicația 1.

Cerințe

SWI-Prolog

A. Vinde AlfaSA calculatoare?

B. Vinde BetaSA televizoare?

C. Ce vinde MagicSRL?

D. Cine vinde calculatoare?

E. Vinde AlfaSA ceva?

F. Vinde cineva accesorii?

G. Cine vinde la fel ca BetaSA un anumit produs?

H. Cine vinde ce?

I. Vinde cineva ceva?

Aplicația 2.

Cerințe

SWI-Prolog

1. Vinde Andrei lalele?

2. Cine vinde crizanteme?

3. Ce vinde Ana?

4. Vinde cineva crini?

5. Vinde Andu ceva?

6. Cine vinde la fel ca Alex un anumit produs?

7. Ce vinde Andu și nu este floare?

Prelucrări Aritmetice

Aplicația 1.

Aplicația 1'.

Aplicația 2.

Aplicația 3.

Aplicația 4.

Aplicația 5.

Aplicația 1.

Se consideră următoarea bază de cunoștințe cu informații privind anumite firme și produsele distribuite.

- Magic srl vinde calculatoare și accesorii.
- Alfa sa vinde televizoare și calculatoare
- Beta sa vinde accesorii.

Cerințe

SWI-Prolog

```
vinde(magicsrl,calculatoare).  
vinde(magicsrl,accesorii).  
vinde(alfasa,calculatoare).  
vinde(alfasa,televizoare).  
vinde(betasa,accesorii).
```

A. Vinde AlfaSA calculatoare?

```
?- vinde(alfasa,calculatoare).  
true.
```

B. Vinde BetaSA televizoare?

```
?- vinde(betasa,televizoare).  
false.
```

C. Ce vinde MagicSRL?

```
?- vinde(magicsrl,X).  
X = calculatoare ;  
X = accesorii.
```

D. Cine vinde calculatoare?

```
?- vinde(Y,calculatoare).  
Y = magicsrl ;  
Y = alfasa.
```

E. Vinde AlfaSA ceva?

```
?- vinde(alfasa,_).  
true.
```

F. Vinde cineva accesorii?

```
?- vinde(_, accesorii).  
true.
```

G. Cine vinde la fel ca BetaSA un anumit produs?

```
?- vinde(betasa,X), vinde(Y,X), Y='betasa'.  
X = accesorii,  
Y = betasa.
```

H. Cine vinde ce?

```
?- vinde(X,Y).  
X = magicsrl,  
Y = calculatoare ;  
X = magicsrl,  
Y = accesorii ;  
X = alfasa,  
Y = calculatoare ;  
X = alfasa,  
Y = televizoare ;  
X = betasa,  
Y = accesorii.
```

I. Vinde cineva ceva?

```
?- vinde(_,_) .  
true.
```

Aplicația 2.

Se consideră următoarea bază de cunoștințe cu informații privind anumite persoane și produsele distribuite.

- Alex vinde trandafiri, lalele și crizanteme.
- Ana vinde lalele și crini.
- Andu vinde lalele și nuci.

Cerințe

SWI-Prolog

```
vinde(alex, trandafiri).  
vinde(alex, lalele).  
vinde(alex, crizanteme).  
vinde(ana, lalele).  
vinde(ana, crini).  
vinde(andu, lalele).  
vinde(andu, nuci).  
floare(X):- not(X = 'nuci').
```

1. Vinde Andrei lalele?

```
?- vinde(andrei,lalele).  
false.
```

2. Cine vinde crizanteme?

```
?- vinde(X,crizanteme).  
X = alex.
```

3. Ce vinde Ana?

```
?- vinde(ana,Y).  
Y = lalele;  
Y = crini.
```

4. Vinde cineva crini?

```
?- vinde(_,crini).  
true.
```

5. Vinde Andu ceva?

```
?- vinde(andu,_).  
true.
```

6. Cine vinde la fel ca Alex un anumit produs?

```
?- vinde(alex,X), vinde(Y,X), Y\='alex'.  
X = lalele,  
Y = ana ;  
X = lalele,  
Y = andu ;  
false.
```

7. Ce vinde Andu și nu este floare?

```
?- vinde(andu,Y), not(floare(Y)).  
Y = nuci.
```

Prelucrări Aritmetice

Aplicația 1.

Să se calculeze media aritmetică a două valori reale.

SWI-Prolog:

```
media_aritmetica(A,B,M_aritmetica):- M_aritmetica is (A+B)/2.
```

Execuție:

```
?- media_aritmetica(40, 34, Media).  
Media = 37.
```

Aplicația 1'.

Să se calculeze media aritmetică a două valori reale, respectiv, media armonica.

SWI-Prolog:

```
media_aritmetica(A,B,M_aritmetica):- M_aritmetica is (A+B)/2.  
media_armonica(A,B,M_armonica):- A \= 0, B \= 0, M_armonica is 2/(1/A + 1/B).
```

Execuție:

```
?- media_aritmetica(40, 34, Media).  
Media = 37.  
?- media_armonica(40, 34, M).  
M = 36.75675675675676.
```

Aplicația 2.

Să se determine maximul/minimul a două valori reale.

SWI-Prolog:

```
maxim(A,B,A):- A>=B.  
maxim(A,B,B):- A<B.  
minim(A,B,B):- A>=B.  
minim(A,B,A):- A<B.
```

Execuție:

```
?- maxim(40, 34, Max).  
Max = 40 .  
?- minim(40, 34, Min).  
Min = 34 .  
?- minim(21, -4, Min).  
Min = -4 .
```

Aplicația 3.

Să se determine maximul/minimul a trei valori reale.

SWI-Prolog:

```
/* Aplicatia 2 */  
maxim(A,B,A):- A>=B.  
maxim(A,B,B):- A<B.  
minim(A,B,B):- A>=B.  
minim(A,B,A):- A<B.  
/* Aplicatia 3 */  
maxim3(A,B,C,Max3):- maxim(A,B,R), maxim(R,C,Max3).  
minim3(A,B,C,Min3):- minim(A,B,R), minim(R,C,Min3).
```

Execuție:

```
?- maxim(40, 34, 12, Max).
Max = 40 .
?- minim(21, -4, 2, Min).
Min = -4 .
```

Aplicația 4.

Să se determine maximul/minimul a două valori reale. **V2 - Prin utilizarea operatorului CUT !**

Observație: Operatorul **CUT !** are rolul de a exclude anumite ramuri de căutare în spațiul soluțiilor în cadrul procesului de backtracking recursiv.

SWI-Prolog:

```
maxim(A, B, A):- A>=B, !.
maxim(A, B, B).
minim(A, B, B):- A>=B, !.
minim(A, B, A).
```

Execuție:

```
?- maxim(40, 34, Max).
Max = 40 .
?- minim(21, -4, Min).
Min = -4 .
```

Aplicația 5.

Să se determine valoarea funcției:

$$f(x, y) = x + y - 2, \text{ dacă } x > -1, y < 1$$

$$f(x, y) = x - y, \text{ în rest}$$

SWI-Prolog:

```
functie_f(X,Y,Rez):- X>(-1), Y<1, Rez is X+Y-2, !.
functie_f(X,Y,Rez):- Rez is X-Y.
```

Execuție:

```
/* x>-1 & y<1 */
?- functie_f(4,-7,Rez).
Rez = -5.
/* x<-1 & y<1 */
?- functie_f(-3,-7,Rez).
Rez = 4.
/* x>-1 & y>1 */
?- functie_f(1,10,Rez).
Rez = -9.
```

