

# Programare Logică – Laboratorul 7

## I/O Operații cu fișiere

Isabela Drămnesc

November 24, 2010

### Noțiuni

- I/O în Prolog

### Exerciții

#### 1. Exemple de scriere

a) `?- write('azi'), write('_e_soare').`

b) `?- writeq("abc").`  
`[97, 98, 99]`  
`true.`

c) `?- write('abc').`  
`abc`  
`true.`

d) `?- writeq('abc').`  
`abc`  
`true.`

e) `?- write('la_la_la').`  
`la la la`  
`true.`

f) `?- writeq('la_la_la').`  
`'la_la_la'`  
`true.`

g) `?- writeq('azi'), writeq('_e_soare').`

*`/* pt azi nu pune ghilimele */`*

h) `?- put(98).`  
`b`  
`true.`

```

/* put scrie un caracter (reprezentand codul ASCII) */

i) ?- put("99").
ERROR: put/1: Type error: 'character' _expected, _found_ '[57,57]'

j) ?- put('99').
ERROR: put/1: Type error: 'character' _expected, _found_ '99'

k) ?- get(L).
|: force

L = 102

l) ?- get(L).
|: f

L = 102

/* get citește un caracter de pe mediul de intrare */

m) ?- write(today is monday),put(8),write(bye).
today is mondabye
true.

n) ?- put(8).
true.

o) ?- write('hello_'), put(8), write(lala),put(98), put(99).
hellolalabc
true.

p) ?- write(hello),put(32),write(man).
hello man
true.

```

## 2. Exemple de citire

```

a) ?- read(X).
|   hello.   /* ceea ce introduce utilizatorul */

X = hello.
      /* read face unificarea lui X cu ce introduce utilizatorul */

b) ?-read(lala).
|   'azi_ploua'.
false.

c) ?- read(X).
|: today is sunshine.

```

```
X = (today is sunshine).
```

```
d) ?- read(X).  
|: 'azi_e_soare'.
```

```
X = 'azi_e_soare'.
```

```
e) ?- read(X).  
|: "azi_e_soare".
```

```
X = [97, 122, 105, 32, 101, 32, 115, 111, 97|...].
```

### 3. Citire din fișier

Creați un fișier pb.txt, iar apoi interogați în Prolog:

a) in cazul in care in fisierul pb.txt nu e scris nimic

```
?- see('C:\\Documents_and_Settings\\ISABELA\\Desktop\\pb.txt'),  
read(X), seen.
```

```
X = end_of_file.
```

```
?- see('C:\\Documents_and_Settings\\ISABELA\\Desktop\\pb.txt'),  
read(X), read(I), seen.
```

```
X = end_of_file.
```

```
I = end_of_file.
```

b) cand fisierul pb.txt contine

```
4.  
5.  
6.  
7.  
10.
```

```
?- see('C:\\Desktop\\pb.txt'), read(X), read(Y),  
read(Z), read(W), read(O), seen.
```

```
X = 4,  
Y = 5,  
Z = 6,  
W = 7,  
O = 10.
```

### 4. Scriere în fișier

```
a) ?- tell('C:\\Desktop\\pb.txt'),  
write('prima_problema'), nl, write('e_rezolvata'),  
write('a_doua_problema_e_sa_mai_scriu_intr-un_fisier'),  
told.
```

**true.**

b) ?- **tell**( 'C:\\Desktop\\pb.txt' ),  
**write**( 'hello' ),**tab**(5),**write**( 'again' ),**told**.

**true.**

c) ?- **tell**( 'C:\\Desktop\\pb2.txt' ),  
**write**( 'a\_doua\_problema' ),**nl**,  
**write**( 'trebuie\_sa\_scriu\_un\_predicat\_care\_aduna\_cifrele' )  
**dintr-un\_fisier** ),**told**.

**true.**

d) ?- **tell**( 'C:\\Desktop\\An2.pdf' ),  
**write**( 'scrie\_in\_pdf\_dar\_il\_deschid\_cu\_notepad' ),**told**.

**true.**

*/\* deci se poate scrie in orice tip de fisier \*/*

#### **5. Problema: Citiți din fișierul pb.txt elementele și afișați suma lor**

a) afisare suma pe mediul de intrare

exemplu:-**see**( 'C:\\Desktop\\pb.txt' ),  
**read**(X),**read**(Y),**read**(Z),**read**(V),**suma**( [X,Y,Z,V] ,W ),  
**write**( 'suma\_elementelor\_din\_fisierul\_pb\_este\_' ),  
**write**(W),**seen**.

**suma**( [], 0 ).

**suma**( [X|T] ,W):-**suma**(T,S),W **is** X+S.

b) afisati suma elementelor din fisierul pb.txt intr-un alt fisier suma.txt

#### **6.**

Pe mediul de intrare pb.txt aveți câte un număr pe fiecare linie urmat de punct.  
Creați un predicat în Prolog care să ordoneze numerele din fișierul pb.txt și  
afișați într-un fișier sortat.txt numerele ordonate.

#### **7.**

Pe mediul de intrare pb.txt aveți câte un număr pe fiecare linie urmat de punct.  
Creați un predicat în Prolog care face shift la stanga (dacă avem în fișier 4.2.3.1.  
după shift la stanga voi avea 2.3.1.4.)și afișați rezultatul într-un alt fișier schim-  
bare1.txt.

#### **8.**

Pe mediul de intrare pb.txt aveți câte un număr pe fiecare linie urmat de punct.  
Creați un predicat în Prolog care face shift la dreapta (dacă avem în fișier

4.2.3.1. după shift la stanga voi avea 1.4.2.3.)și afișați rezultatul într-un alt fișier schimbare2.txt.

## 9.

Pe mediul de intrare pb.txt aveți câte un număr pe fiecare linie urmat de punct. Creați un predicat în Prolog care să separe numerele din fișierul pb.txt în numere pare și numere impare. Numerele pare vor fi afișate într-un fișier pare.txt, iar cele impare vor fi afișate într-un fișier impare.txt.

## Tema:

Din [tema 3](#) problemele 1,7,8,9,13,14,15,16,17,19,24,25 utilizând fișiere. Adică citiți din fișier, iar rezultatul îl afișați în alt fișier.

Într-un fișier aveți 30 de linii. Fiecare linie conține câte un număr urmat de punct. Afișați într-un alt fișier numerele în ordine crescătoare!