

## Laboratorul 9: Exercitii suplimentare/facultative

### Exercițiul 1

- a) Scrieți o funcție care numără câte propoziții sunt într-un text dat. Puteti scrie o funcție auxiliară `sfChr` care verifică dacă un caracter e sfârșit de propoziție. Considerăm semne de sfarsit de propozitie: punct '.', semnul intrebării '?', semnul exclamării '!', doua puncte ':'. In rezolvarea exercitiului puteti folosi doar recursie si `sfChr`, fara metoda prin selectie sau functii de nivel inalt.
- b) Rezolvati același exercițiu folosind doar metoda prin selectie.

### Exercițiul 2

Scrieți o funcție `liniiN` care are ca parametru o matrice de numere întregi (`[[Int]]`) și un număr întreg `n`, și verifică dacă toate liniile de lungime `n` din matrice au doar elemente strict pozitive. In rezolvarea exercitiului folositi functii de nivel inalt.

### Exercițiul 3

Se dau următoarele tipuri de date ce reprezinta puncte cu numar variabil de coordonate intregi:

```
data Punct = Pt [Int]
           deriving Show
```

Arbori cu informația în frunze și clasă de tipuri `ToFromArb`

```
data Arb = Vid | F Int | N Arb Arb
           deriving Show
```

```
class ToFromArb a where
  toArb :: a -> Arb
  fromArb :: Arb -> a
```

Să se scrie o instanță a clasei `ToFromArb` pentru tipul de date `Punct` astfel incat lista coordonatelor punctului sa coincidă cu frontiera arborelui.

```
toArb (Pt [1,2,3])
  N (F 1) (N (F 2) (N (F 3) Vid))
fromArb $ N (F 1) (N (F 2) (N (F 3) Vid)) :: Punct
  Pt [1,2,3]
```