

[Tablou de bord](#) / [Cursurile mele](#) / [2020-2021](#) / [Facultatea de Matematică și Informatică](#) / [Studii universitare de licență](#)
/ [Informatică](#) / [Limba română](#) / [Forma de învățământ - IF](#) / [Anul 2](#) / [Semestrul I](#) / [Programare functionala - Seria 23](#)
/ [Organizarea online a cursurilor/laboratoarelor](#) / [Examen](#)

Început pe	luni, 1 februarie 2021, 10:10
Status	Finalizat
Completat pe	luni, 1 februarie 2021, 12:10
Timp de parcurs test	1 oră 59 min
Notă obținută	Nu este încă notat

1 întrebare

Complet

Fără notă

IMPORTANT!

Denumirile tuturor funcțiilor din rezolvările dvs. trebuie să înceapă cu litera 'f', apoi două cifre, apoi inițială nume și inițială prenume, și apoi alte litere, dacă doriți

de exemplu, pentru numele Șerbănuță Traian, un nume valid de funcție este **f13stFunctieBuna**

Scrieți în caseta de răspuns un astfel de identificator.

f09cnFunctie

2 întrebare

Complet

Marcat din 30,00

Se da urmatorul tip de date reprezentand enciclopedii

```
type Concept = String
type Definitie = String
type Categorie = String
data Enciclopedie = Intrare Concept Definitie | Lista Categorie [Enciclopedie]
                    deriving Show
```

a) Sa se scrie o functie care numara cate cuvinte reprezentand concepte diferite sunt intr-o enciclopedie data ca parametru (in scrierea conceptelor nu se face distinctie intre litere mari si mici, de exemplu "elefant" si "Elefant" reprezinta acelasi concept; sunt considerate concepte numai cuvintele care apar intr-un tip de date Intrare Concept Definitie in pozitia Concept).

Exemplu:

```
enc1 = Lista "animal"[Lista "mamifer"[Intrare "elefant" "acesta e un elefant", Intrare "caine" "acesta este un caine", Intrare "pisica" "aceasta este o pisica"], Intrare "zebra" "zebra este un animal"]
enc2 = Lista "Animal"[Lista "animal" [Intrare "Elefant" "acesta e un elefant", Intrare "caIne" "acesta este un caine"], Intrare "caine" "definitie"]
```

enc1 are 4 concepte ("elefant", "caine", "pisica", "zebra") iar enc2 are o 2 concepte ("Elefant" si "caine")

b) Sa se faca instanta a clasei Eq pentru tipul de date Enciclopedie, astfel incat doua enciclopedii se considera egale daca contin aceleasi concepte distincte indiferent de ordinea in care sunt scrise si de gruparea lor in categorii (in scrierea conceptelor nu se face distinctie intre litere mari si mici; sunt considerate concepte numai cuvintele care apar intr-un tip de date Intrare Concept Definitie in pozitia Concept). Exemplu:

```
enc1 == enc 2 = False
```

c)

```
type Dictionar = [(Concept, [Definitie])]
```

Scriti o functie care transforma o valoare de tip Enciclopedie intr-o valoare de tip Dictionar astfel incat unei valori de tip Concept i se asociaza lista definitiilor, ignorand clasificarea in categorii. In scrierea conceptelor nu se face distinctie intre litere mari si mici, de exemplu "elefant" si "eLEfant" reprezinta acelasi concept, in Dictionar va apare numai una din scrieri, la alegerea programatorului. Nu se impun restrictii asupra ordinii definitiilor asociate aceluasi concept.

De exemplu,

```
enc3 = Lista "animal"[Lista "mamifer"[Intrare "elefant" "def1", Intrare "caine" "def2"], Intrare "eLEfant" "def3"]
```

poate fi transformata in

```
[("elefant", ["def1", "def3"]), ("caine", "def2")]
```

```
--a
f09cnParcure (Intrare c desc) = lower c
f09cnParcure (Lista cat (h:t)) = f09cnParcure h

-- f09cnParcure (Intrare c desc) = [lower c]
--f09cnParcure (Lista cat (h:t)) = (f09cnParcure h) ++ (f09cnParcure t)
--vreau sa parcure fiecare intrare si sa le adun pe toate intr-o lista noua
-- nu stiu cum sa parcure lista de intrari, ma opresc la prima intrare

--b
f09cnLowerLista lst = [lower x | x<-lst]
Instance Eq Enciclopedie where
  a == b = let pairs = zip (sort (f09cnLowerLista (f09cnParcure a))) (sort(f09cnLowerLista (f09cn
    sum [1| x<-pairs, (fst x) == (snd x)] == length (f09cnParcure a)
-- presupunand ca merge functia de la a, as sorta output-ul fiecărei liste de categorii (dupa ce dau
-- apoi as da zip si verifica daca perechile dau match
-- daca nu dau => avem diferenta, deci nr de perechi matching != lungimea unei liste
```

3 întrebare

Complet

Marcat din 30,00

[