

Exerciții - Săptămâna 4

Formatori:

Tutor: [Militaru Mihai-Adrian](#)  

Tutor: [Dragomir Titian-Cornel](#)  

+8

Data de începere a cursului:

 25.09.2023

 [Utilizatori înscrși](#)

 [Calendar](#)

 [Note](#)

 [Cursurile mele](#) [S1-L-AC-CTIRO1-LSD](#) [Săptămâna 4: Liste](#) [Exerciții - Săptămâna 4](#)

Exerciții - Săptămâna 4

Construirea de liste

Partea I

- Fiind dată o listă cu numere naturale, afișați toate elementele pare.
- Fiind dată o listă cu șiruri de caractere, afișați toate elementele care au mai puțin de 3 caractere.
- Se dă o listă cu mai multe tipuri de date (întregi, reale, șiruri de caractere, valori booleene). Să se sorteze lista crescător, apoi descrescător. (explicatie: la rulare va da eroare pentru ca nu se pot sorta liste cu elemente de diferite tipuri. nu se poate compara numar<sir de caractere. Se pot sorta doar liste care contin acelasi tip de elemente)
- Să se sorteze o listă care conține numere întregi în funcție de ultima cifră a numărului (în ordine crescătoare a ultimei cifre din număr).
- Să se înmulțească toate elementele unei liste de numere reale. Se va folosi funcția reduce().
- Să se creeze o nouă listă care să conțină toate elementele unei liste inițiale la puterea a 3-a. Se va folosi funcția map().
- Să se creeze o nouă listă care conține doar elementele numere prime dintr-o listă inițială. Se va folosi funcția filter().

Partea a II-a

- a) Implementați funcții care construiesc lista cifrelor unui număr care satisfac o condiție anume (cifre impare, pare, mai mici decât 7, etc. la alegere), în ordine normală și inversă
b) Implementați o funcție care construiește lista cifrelor unui număr care satisfac o condiție dată ca parametru sub forma unei funcții cu tipul `int -> bool`.
c) Invers, dată fiind o listă de cifre, construiți numărul format doar din cifrele care respectă o condiție (dată ca parametru funcție cu tipul `int -> bool`). Rezolvați problema direct, recursiv, și apoi prin folosirea lui `filter` (selectarea cifrelor) cu `reduce` (pentru construirea numărului).
- Implementați funcția `fromto` care generează lista numerelor întregi dintr-un interval dat, scrieți o funcție care creează lista tuturor întregilor dintr-un interval dat, divizibili cu o valoare dată `d`.
Indicație: Găsiți cel mai mare număr divizibil din interval, și continuați pas cu pas.
- a) Implementați funcția `nth` care returnează al n-lea element dintr-o listă.
b) Implementați o funcție `firstn` care returnează o listă cu primele n elemente dintr-o listă dată.

Parcurgeri de liste

- a) Implementați o funcție numită `filter`, cu același comportament ca și funcția predefinită `filter`, folosind `reduce`.
b) Implementați funcția `exists` care determină (returnează adevărat/fals) dacă există un element din listă care satisface o condiție (o funcție de

element cu valoare booleană, dată ca parametru).

6. a) Implementați folosind `reduce` o funcție `countIf` care ia ca parametru o funcție `f` cu valori boolene și o listă și returnează numărul de elemente pentru care funcția `f` e adevărată.

b) Implementați similar o funcție `sumIf` care calculează suma tuturor elementelor (presupuse întregi) pentru care funcția `f` e adevărată.

7. Implementați funcțiile `split` și `combine` care transformă o listă de perechi într-o pereche de liste, și invers.

Ex: `split([(1,2), (3,4), 5,6)]) -> ([1,3,5], [2,4,6])`

`combine([1,3,5], [2,4,6]) -> [(1,2), (3,4), 5,6]`

8. Implementați funcția `partition` care ia ca parametru o funcție cu valori boolene și o listă și returnează o pereche de liste, cu elementele care satisfac, respectiv nu satisfac funcția `f`.

`partition (lambda x : x >= 5) [4,6,7,5,4,8,9] -> ([6, 7, 5, 8, 9], [4, 4])`

9. Scrieți o funcție care ia o listă de cifre și returnează valoarea numărului cu cifrele respective.

10. Scrieți o funcție care *elimină duplicatele* consecutive: ia ca parametru o listă și construiește o listă în care toate secvențele de elemente consecutive egale au fost înlocuite cu un singur element.

11. Scrieți o funcție care *compară* două liste după următoarea relație de ordine: o listă mai scurtă e "mai mică" decât una mai lungă; dacă lungimile sunt egale, ordonarea e determinată de prima pereche de elemente diferite. Evitați parcurgerea inutilă sau repetată a listelor. Funcția va returna un întreg negativ, 0 sau pozitiv în funcție de ordonarea celor două liste argument.

Sortarea prin interclasare

12. Scrieți o funcție care *interclasează* două liste, fiecare ordonată crescător, adică returnează lista cu elementele din ambele liste, ordonate. Comparați primele elemente din ambele liste pentru a decide care va fi primul în rezultat, și apoi continuați cu listele rămase.

13. Scrieți o funcție care *desparte* o listă în două liste a căror lungime diferă cu cel mult 1, punând alternativ câte un element în fiecare din liste. (Funcția va returna o *pereche* de liste).

14. *Sortați* o listă prin interclasare (*mergesort*), despărțind lista în două jumătăți, sortând recursiv fiecare din ele, și apoi interclasând cele două liste sortate. Folosiți funcțiile din cele două probleme anterioare.

◀ Laboratorul 4

Sari la...

Laborator 4 - Notebook (Subgrupele 1.1C, 1.2D) ▶

✉ Contactați serviciul de asistență

Sunteți conectat în calitate de Ciobanu Daria-Andreea (Delogare)

S1-L-AC-CTIRO1-LSD

Meniul meu

Profil

Preferinte

Calendar

👤 ZOOM

Română (ro)

English (en)

Română (ro)

Rezumatul păstrării datelor

Politici utilizare site