



Demonstrature 9 - dodatni zadaci

Zadatak 1. Implementirajte funkciju *sumFoldl* koja prima cijeli broj n i listu cijelih brojeva xs te vraća zbroj svih članova dane liste i broja n , ali na način da se rezultat svakog poziva funkcije sprema u n .

Npr. `sumFoldl 4 [7,5,1,2] = sumFoldl 11 [5,1,2] = ... = 19`

Zadatak 2. Implementirajte funkciju *sumScanl* koja radi isto što i funkcija *sumFoldl*, ali sprema rezultat svakog poziva funkcije u listu. Provjerite čemu služi funkcija *scanl*.

Npr. za listu `[7,5,1,2]` i $n = 4$, funkcija treba vratiti `[4,11,16,17,19]`.

Zadatak 3. Implementirajte funkciju *bubbleSort* koja prima listu podataka iz klase *Ord* te ih sortira od najmanjeg prema najvećem koristeći algoritam *Bubble sort*. Preporučeno je korištenje funkcije *foldr* i pomoćne funkcije *bubble* koja radi jedan prolazak kroz podlistu prvih k članova dane liste. Iskoristite svojstvo funkcije *bubble* koje nam osigurava da se nakon jednog prolaska kroz neku podlistu najveći element te podliste nalazi na njenom desnom kraju.

Zadatak 4. Implementirajte funkciju *opseg* koja prima listu točaka (točke su zadane kao uređeni parovi realnih brojeva) te računa opseg mnogokuta kojem pripadaju dane točke. Pokušajte riješiti ovaj zadatak koristeći funkciju *foldr* ili *scanr*.

Zadatak 5. Implementirajte funkciju *insertElem* koja prima podatak n tipa *a* i sortiranu listu tipa *a* te ubacuje n u listu, ali tako da lista ostane sortirana.

Zadatak 6. Implementirajte funkciju *insertionSort* koja sortira listu koristeći algoritam *Insertion sort*. Provjerite čemu služi funkcija *flip*.