

Demonstrature 8 - dodatni zadaci

Napomena 1. Ako u zadatku nije drugačije navedeno, onda nije potrebno provjeravati ispravnost inputa.

Zadatak 1. Implementirajte funkciju *traceFor* koja računa trag realne (kvadratne) matrice koristeći "for petlju" te funkciju *trace* koja radi isto što i funkcija *traceFor*, ali pomoću generatora liste ili funkcije *map*.

Zadatak 2. Implementirajte funkciju *forSum* koja prima cjelobrojnu kvadratnu matricu A reda n s elementima a_{ij} te računa sumu

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} f(a_{ij}), \quad f(a_{ij}) = \begin{cases} i \cdot j \cdot a_{ij}, & i = j \\ 1, & i \neq j, 3 \nmid a_{ij} \\ 0, & \text{inače} \end{cases}$$

koristeći "for petlju" ili funkcije višeg reda. Primijetite da indeksi idu od 0 do $n - 1$.

Zadatak 3. Implementirajte funkciju *matSum* koja prima dvije realne matrice jednakih dimenzija i vraća njihov zbroj.

Zadatak 4. Po uzoru na funkciju *trace*, implementirajte funkciju *transpose* koja prima realnu matricu i transponira ju.

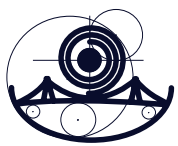
Zadatak 5. Implementirajte funkciju *matDotVec* koja prima matricu A i vektor v , a vraća rezultat množenja $A \cdot v$. Ako retke matrice A zamislimo kao vektore (a_1, \dots, a_m) , onda je rezultat množenja $A \cdot v$ vektor s komponentama $(\langle a_1, v \rangle, \dots, \langle a_m, v \rangle)$.

Zadatak 6. Implementirajte funkciju *matMul* koja prima dvije ulančane matrice, A i B , te računa njihov produkt. Ako stupce matrice B zamislimo kao vektore $[b_1, \dots, b_r]$, tada je rezultat množenja $A \cdot B$ matrica čiji su stupci vektori $[A \cdot b_1, \dots, A \cdot b_r]$.

Zadatak 7. Implementirajte funkciju *listToInt* koja prima listu znamenaka te vraća prirodan broj s istim znamenkama. Npr. za listu $[1, 8, 2, 5]$ funkcija treba vratiti rezultat 1825. Ukoliko lista sadrži više od 18 elemenata ili je prazna, vratite -1.

Zadatak 8. Koristeći funkciju *foldl* (ili *foldr*) i funkciju *zip*, implementirajte funkciju *binom* koja prima redak iz Pascalovog trokuta reprezentiran listom cijelih brojeva, a vraća sljedeći redak.

Zadatak 9. Koristeći funkcije *binom* i *foldl* (ili *foldr*), implementirajte funkciju *pascal* koja prima $n \in \mathbb{N}_0$ te računa n -ti redak Pascalovog trokuta. Pretpostavite da je nulti redak Pascalovog trokuta jednak $[1]$.



Zadatak 10. 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 11. 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*.

Zadatak 12. Implementirajte funkciju *insert* koja prima podatak n tipa *a* i sortiranu listu tipa *a* te ubacuje n u listu, ali tako da lista ostane sortirana.

Zadatak 13. Implementirajte funkciju *insertionSort* koja sortira listu koristeći algoritam *Insertion sort*.

Zadatak 14. Implementirajte funkciju *maxlen* koja prima dvije liste istog tipa te vraća onu koja je dulja. Ukoliko su liste iste duljine, funkcija vraća drugu.

Zadatak 15. Implementirajte funkciju *lss* koja prima niz elemenata tipa *a* i vraća njegov najveći sortirani podniz. Ukoliko je više takvih, vratite onog koji se prvi pojavio.