



Demonstrature 4 - dodatni zadaci

Zadatak 1. Implementirajte funkciju koja prima nepraznu konačnu listu te vraća uređeni par (a, b) , gdje je a prvi element liste, a b zadnji element liste.

Zadatak 2. Implementirajte funkciju koja prima listu x te vraća listu parova uzastopnih elemenata od x . Npr. ako lista sadrži elemente $[a, b, c, d]$, funkcija treba vratiti $[(a, b), (b, c), (c, d)]$.

Zadatak 3. Implementirajte funkciju koja prima listu tipa `Double`, a vraća listu korijenjena njenih elemenata s parnim indeksima. Pretpostavite da indeksi kreću od 0.

Zadatak 4. Implementirajte funkciju koja prima listu proizvoljnog tipa te vraća njen predzadnji element. Pretpostavite da lista sadrži barem dva elementa.

Zadatak 5. Implementirajte funkciju `is_elem` koja prima argument x iz klase `Eq` te konačnu listu S koja sadrži elemente istog tipa kao x . Ukoliko se x nalazi u listi S , funkcija vraća `True`, a u suprotnom vraća `False`.

Zadatak 6. Implementirajte funkciju koja prima string koji se sastoji isključivo od malih slova engleske abecede i razmaka te iz njega izbacuje sve samoglasnike. Prvo implementirajte koristeći pattern matching, a zatim koristeći čuvare i funkciju iz prethodnog zadatka.

Zadatak 7. Implementirajte funkciju `is_leap` koja provjerava je li godina prijestupna. Po definiciji, godina je prijestupna ako je ili djeljiva sa 400 ili je djeljiva sa 4, a da pri tom nije djeljiva sa 100.

Zadatak 8. Implementirajte funkciju `binom` koja prima redak iz Pascalovog trokuta reprezentiran listom cijelih brojeva, a vraća sljedeći redak.

Zadatak 9. (Zadatak s drugih vježbi) Implementirajte funkciju `apply` koja:

- prima funkciju f , element a proizvoljnog tipa te $n \in \mathbb{N}_0$;
- n puta primjenjuje funkciju f na a ;
- vraća rezultat n -te primjene.

Npr. za $n = 3$ funkcija `apply` treba vraćati $f(f(f(a)))$.

Zadatak 10. Koristeći funkcije `binom` i `apply` iz prethodnih dvaju zadataka, 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 11. 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 `apply` i pomoćne funkcije koja radi jedan prolazak kroz listu.