

Zadatak 1. Implementirajte funkciju *fast_pow* koja prima broj x i prirodan broj n te računa x^n sljedećom formulom.

$$x^n = \begin{cases} \left(x^{\frac{n}{2}}\right) \left(x^{\frac{n}{2}}\right) & \text{paran } n \\ x \left(x^{n-1}\right) & \text{neparan } n \end{cases}$$

Zadatak 2. Napišite funkciju koja pomoću uzorka "_" i generatora liste računa duljinu proizvoljne liste.

Zadatak 3. Napišite četiri funkcije koje iz uređenih četvorki vraćaju redom prvi, drugi, treći i četvrti element. Prilikom implementacije koristite uzorak "_".

Primjer 1.1.

fir(1,2,3,4) == 1

sec(1,2,3,4) == 2

thi(1,2,3,4) == 3

fou(1,2,3,4) == 4

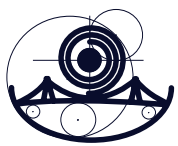
Zadatak 4. Napišite funkciju koja za listu uređenih parova vrati listu prvih elemenata svakoga para.

Zadatak 5. Koristeći čuvare, napišite rekursivnu funkciju koja vraća -ti element liste u obliku jednočlane liste. Ukoliko -ti element nije moguće pronaći, vratite praznu listu.

Zadatak 6. Napišite program koji na ulazu prima dvije točke $P, Q \in \mathbb{R}^3$ i vraća njihovu euklidsku udaljenost.

Zadatak 7. Napišite program koji računa skalarni produkt dva konačna realna vektora reprezentirana nepraznim listama iste duljine. Skalarni produkt vektora računamo tako da pomnožimo odgovarajuće komponente vektora (s istim indeksima), te tako dobivene rezultate zbrojimo. Npr. $\langle [1,2], [3,4] \rangle = 1 \cdot 3 + 2 \cdot 4$.

Zadatak 8. Implementirajte funkciju *count_all* koja prima `String` i vraća `[(Char, Int)]` koliko se koje slovo puta pojavljuje u danom stringu.



Zadatak 9. Napišite program koji iz unešenog stringa koji se sastoji isključivo od malih slova engleske abecede, brojeva i razmaka izbacuje samoglasnike i parne znamenke.

Zadatak 10. Napišite program koji na ulazu prima listu L i element istog tipa e . Program treba iz L izbaciti točno dvije kopije elementa e . Ukoliko se u listi L nalazi manje od dvije kopije od e ne činite ništa.

Zadatak 11. Implementirajte funkciju koja paran prirodan n broj preslika u $\frac{n}{2}$, neparan u $-\frac{n-1}{2}$, dok sve cijele brojeve manje od 1 preslika u nulu.

Zadatak 12. Napišite funkciju koja na svaki element unešene liste cijelih brojeva primjenjuje funkciju iz prethodnog zadatka. Pogledajte čemu služi funkcija `map`.

Zadatak 13. Implementirajte funkciju `i2b` koja prirodan broj n zapiše kao string nula i jedinica u binarnom zapisu. Npr. broj 9 ima binarni zapis 1001, pa funkcija `i2b` vraća string "1001".

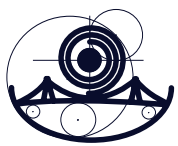
Implementirajte funkciju `bplus` koja prima dva binarna stringa koji reprezentiraju prirodne brojeve u binarnom zapisu. Funkcija vraća sumu dva prirodna broja u obliku binarnog stringa.

Primjer 1.2. Na inputu se nalaze brojevi 4 i 1. Funkcija `i2b` brojeve pretvara u binarne stringove "100" i "1". Potom, funkcija `sumb` zbraja dva binarna stringa i vraća rezultat "101".

Napomena: Binarne stringove sumiramo tako da ih prvo svedemo na binarne stringove iste duljine. To možemo napraviti tako da kraći binarni string nadopunimo nulama s lijeva. Tako ćemo binarne stringove "100" i "1" sumirati tako da "1" nadopunimo do "001". Dalje pratite postupak zbrajanja u bazi 2.

Zadatak 14. Neka je u matrici znakova xo (lista lista tipa `Char`) dano stanje ploče u igri križić kružić na kraju partije (redak po redak). Implementirajte funkciju `checkxo` koja provjerava ima li u danom polju pobjednika. Ako je pobjednik igrač X ispišite "X". Ako je pobjednik igrač O ispišite "O". Inače, ispišite "N" (neriješeno).

Podsjetnik: Križić kružić se igra na ploči dimenzije 3×3 . Igrači naizmjenice u polja upisuju svoj znak (X ili O). Pobjednik je igrač koji u redak, stupac ili dijagonalu upiše svoja tri znaka.



Primjer 1.3. Ako je

$$xo = ["XXX", "XOO", "OOX"]$$

program ispisuje "X". Prvi redak matrice je "XXX", drugi "XOO" i treći "OOX".

Zadatak 15. Implementirajte funkciju koja enkodira niz znakova sastavljen isključivo od malih slova engleske abecede koristeći se znakovima '.' i '-', tako da je:

- a == "...."
- b == ".... -"
- c == "... -."
- d == "... - -"
- e == ".. - .."
- f == ".. - . -"
- g == ".. - -."
- h == ".. - - -"
- itd.

Između koda za svaki znak stavite '—'. Također, na početak i kraj riječi stavite znak "—".

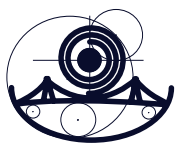
Napomena: U Haskell biblioteci Data.Char nalaze se funkcije *chr* i *ord*. Funkcija *ord* za *Char* na ulazu vraća njegovu ASCII vrijednost. Funkcija *chr* za ASCII vrijednost (*Int*) na ulazu vraća odgovarajući *Char*. Biblioteku uvrštavamo u naš program naredbom `import`.

Primjer 1.4. Input: "hello"

Output "|.. - - - |.. - ..|. - . - -|. - . - -|. - - - .|"

Zadatak 16. Neka je dana gredica sa suncokretima u kojoj je n suncokreta visine h_i , $i = 1, \dots, n$ poredano jedan po jedan u liniji. Gredica je reprezentirana nizom cijelih brojeva kojemu svaki element predstavlja visinu suncokreta.

Vrtljar koji se brine o gredici posjeduje posebnu UV svjetiljku koja može potaknuti obasjane suncokrete na rast za jednu jedinicu visine tokom jednog dana. Svjetiljku je moguće postaviti isključivo iznad jednog od suncokreta. Suncokret smatramo obasjanim ukoliko se između njega i svjetiljke ne nalazi drugi, viši suncokret koji mu stvara sjenu.



Implementirajte funkciju *day* koja na ulazu prima cijeli broj x i gredicu g . Funkcija simulira prolazak jednog dana pod UV svjetlom koje se nalazi na poziciji x te povećava visinu svih obasjanih suncokreta za 1.

Napomena: Indeksi u nizu počinju s nulom.

Primjer 1.5. Input 3 [1, 4, 2, 1, 3, 1, 7, 1]
Output [1, 5, 3, 2, 4, 1, 8, 1]