

Cvícenie 10

Instrukcie: Vyrieste všetky ulohy. Skúste čo najviac z nich spraviť už na cvícení. Ak nejakú úlohu nestihnete na cvícení a nebudete ju vedieť vyriešiť ani doma, opýtajte sa cvičiaceho na ďalšom cvícení.

1. Definujte funkciu, ktorá bude načítavať čísla z klavesnice, kým užívateľ nezadá 0 a ktorá vráti také číslo, ktoré bolo načítané najviackrát. (Ak napríklad boli čísla 3 a 105 načítané 4-krát a všetky ostatné čísla boli načítané menejkrát, funkcia vráti jedno z čísel 3 a 105.)
2. Vyrieste úlohu 10.4 zo sekcie 10.15 v knihe. Pozor! Funkcia `chop` má zmeniť vstupný zoznam, nie vytvoriť nový!
3. Uvažujme zoznam `t`. Zoznam `t` bude mať `n` prvkov. Prvky zoznamu `t` budú zoznamy obsahujúce `k` čísel (každý z `n` zoznamov obsahuje `k` čísel). Môžeme to potom brať tak, že zoznam `t` reprezentuje maticu rozmeru $n \times k$, kde každý z prvkov zoznamu `t` reprezentuje jeden riadok matice. Definujte funkciu, ktorej argumentami budú dve matice `A` a `B`. Funkcia vráti súčet `A+B`. V prípade, že matice nie je možné sčítať, funkcia vypíše “Zadane matice nie je možné sčítať”.
4. Definujte funkciu, ktorej argumentami budú dve matice `A` a `B` (matice budú reprezentované rovnako ako v úlohe 2). Funkcia vráti súčin `A*B`. V prípade, že matice nie je možné vynásobiť, funkcia vypíše “Zadane matice nie je možné vynásobiť”.
5. Definujte funkciu, s nasledovnými vlastnosťami. Argumentom funkcie bude zoznam `t`. Prvky zoznamu `t` budú zoznamy obsahujúce dve čísla. Nech `[a, b]` je prvok zoznamu `t`. Zoznam `[a,b]` bude reprezentovať otvorený interval od čísla `a` po číslo `b` (budeme predpokladať, že v každom z vnútorných zoznamov je číslo na 0-tom indexe menšie ako číslo na 1-vom indexe). To znamená, že zoznam `t` budeme chápať, ako zoznam intervalov. Funkcia vráti hodnotu `True`, ak je spoločný prienik všetkých intervalov v zozname `t` neprázdny. Inak vráti funkcia hodnotu `False`.
6. Vyrieste úlohy 10.7 – 10.12 zo sekcie 10.15 v knihe.
Poznámka 1: V úlohe 10.10 sa zoznamíte s veľmi užitočnou metódou zvanou “binary search”. Skúste túto úlohu vyriešiť bez použitia modulu `bisect`. Keď budete mať úlohu 10.10 hotovú, určite si pozrite aj riešenie v knihe!
Poznámka 2: V úlohe 10.12 by ste určite **nemali** skúšať všetky možné pary slov. To by trvalo príliš dlho. Treba vymyslieť iný efektívnejší spôsob ako túto úlohu riešiť.