

12. óra

Kötelező feladatok

- Hozzunk létre egy új típust `Student` néven, mellyel hallgatókat fogunk reprezentálni.
 - Tartalmazzon azonosítót, átlagot és kort.
 - Az átlag típusa legyen kétszeresen pontos lebegőpontos szám, a többi egész szám.
- Vizsgáljuk meg a memóriaszükségletét a típusnak, kísérletezzünk az adattagok sorrendjével, és figyeljük meg ennek hatását.
- A praktikusság jegyében hozzunk létre egy alias típust `typedef` fel, hogy a típust tudjuk a `struct` előtag nélkül is használni.
- Hozzunk létre egy függvényt mely egy `Student` tömböt kap és visszaadja a legmagasabb átlagú hallgatónak az azonosítóját. Kezeljük külön a BSc-s, MSc-s és PhD-s hallgatókat, erre hozzunk létre egy felsoroló típust (`enum`) `Type` néven és adjuk hozzá a `Student`hez ad: A hallgató típusától függően tároljunk különböző plusz adatokat egy `union` típus segítségével.
 - BSc: összesen hallgatott kurzusok száma(`int`).
 - MSc: az összesített korrigált kreditindex(`double`)
 - PhD: legmagasabb impaktfaktorú folyóiratnak az impaktfaktora, melybe publikált (`double`), illetve Erdős-számát (`int`) egy `struct` segítségével.
- Vizsgáljuk meg a típus memóriaszükségletét és vessük össze, mi lett volna, ha `struct`ot használunk helyette.
- Írjunk egy függvényt(`student_init`) mely egy `Type` paramétert kapva, létrehoz a heapen egy `Student` példányt a megfelelő `Type`-al
 - a megfelelő adattagokat feltölti random adatokkal, majd visszaad egy mutatót a példányra.
- Töltsünk fel egy tömböt ilyen `Student` példányokra mutatókkal, majd módosítsuk a 3. feladatban írt függvényt:
 - azonosító helyett a példányra mutatót adjon vissza. Ügyeljünk a memóriaszivárgás elkerülésére.

Opcionális feladatok

- A korábbi órán megírt mátrixszorzáshoz, hozzunk létre egy `Matrix` típust,
 - mely tartalmazza a mátrix dimenzióinak méretét, illetve a tömb első elemére mutatót adattagként.
- Az indexelés könnyítése érdekében írjunk egy `at` függvényt, mely vár egy mátrix példányra mutatót, sor és oszlop indexet,
 - ezek alapján visszatér a indexek által jelölt elemre mutatóval.
- Alakítsuk át a mátrixszorzást ezek segítségével.

Haladó feladatok

- Hozzunk létre egy egészeket tartalmazó dinamikus tömb típust, mely adattagként tárolja a tömb aktuális méretét, kapacitását és egy tömbre mutatót
- Hozzunk létre egy függvényt mely a paraméterül kapott kezdeti kapacitás alapján létrehoz egy új példányt, megfelelően beállított adattagokkal.
 - Majd ennek párját, mely megfelelően felszabadítja a kapott példányt.
- Írjunk egy függvényt mellyel hozzá tudunk adni egy elemet a tömb végére. Ha megtelt a tömb, foglaljunk le egy kétszer nagyobb tömböt.
 - Majd a korábbi elemeket másoljuk át és frissítsük az adattagokat. Ügyeljünk a memóriaszivárgásra.