

13. óra

Kötelező feladatok

1. Valósítsunk meg egy olyan verem adatszerkezetet, amiben "dobozokat fogunk tudni egymásra pakolni". Ehhez hozzunk létre egy `Box` nevű struktúrát, ami rendelkezik egy egész szám típusú súly értékkel (`weight`) és egy Box struktúrára mutató pointerrel (`next`).
2. Hozzunk létre egy globális pointer változót a Box struktúrából (`top`). Írjuk meg az `initialize` függvényt, amely NULLra állítja a top változót. A main függvényben hívjuk meg az initialize-t!
3. Írjuk meg az `isEmpty` függvényt, amely visszaadja, hogy üres-e a stack vagy sem. (A stack akkor üres, ha a top változó NULL pointer.)
4. Írjuk meg a `peek` függvényt, ami a jelenlegi legfelső Box súlyával tér vissza. (Ne feledkezzünk meg arról az esetről, amikor a stack üres!)
5. Írjuk meg a `push` függvényt, ami egy egész számot kap paraméterül, ez lesz a következő doboz súlya. Hozzunk létre egy új elemet a stacken, és állítsuk be a súlyát. Az új elem next adattagja az előző adat tagra mutasson! A top változó mutasson az új elemre.
6. Írjuk meg a `pop` függvényt, ami felszabadítja a verem tetején lévő elemet, és a top változót az egyel alatta lévőre állítja. (Ne feledkezzünk meg az üres stack lehetőségéről sem!)
7. Írjuk meg a `copyTop` függvényt, amely lemásolja a verem legfelső elemét, és a verem tetejére helyezi a másolatot. Nézzük meg, mi történik, ha csak a legfelső elemre mutató pointert másoljuk le, illetve ha új memóriaterületet allokalunk a másolatnak, és adattagonkénti másolást végzünk!

Opcionális feladatok

1. Készítsünk el egy `Person` nevű struktúrát, amely tartalmaz (legalább) egy név és egy életkor értéket. A nevet egy char[30] ban, az életkort pedig egy int* által hivatkozott memóriaterületen tároljuk el.
1. A struktúra kényelmes használata érdekében alkalmazzunk típus alias-ozást.
2. A main függvényen belül hozzunk létre egy Persont, melynek beállítjuk az adatait. Hozzunk létre egy újabb Persont, ami értékül kapja az első Persont. (p1: p2 = p1;)
3. Írassuk ki a standard outputra a két Person értékeit. Módosítsuk az első Person életkorát. Ismét írjuk ki a két Person adatait. Miért változik meg a második Person életkorának adata? (shallow copy)
4. Javítsuk ki a programunkat úgy, hogy a Personok közötti értékadás helyett a konkrét mutatott értékeket másoljuk ki (deep copy). Ismét futtassuk le a programot, és demonstráljuk, hogy az első Person életkorának adatainak módosításával már nem változik meg a második Person életkora.