

*Az alábbi feladatok típusait egy-egy osztály segítségével valósítsa meg. Egy osztály szolgáltatásainak (metódusainak) bemutatásához olyan főprogramot kell készíteni, amelyik egy menü segítségével teszi lehetővé a metódusok tetszőleges sorrendben történő kipróbálását. A főprogram ezért példányosítson legalább egy objektumot, amelyre a menüpontok közvetítésével lehet meghívni az egyes metódusokat. Természetesen szükség lesz minden tevékenység után az objektum állapotának kiírására vagy egy az objektum állapotát kiíró külön menüpontra. Azoknál az osztályoknál, ahol vannak olyan metódusok (esetleg barát függvények), amelyek több objektum közötti műveleteket valósítanak meg, a főprogram több objektum létrehozására és azok állapotának kiírására adjon lehetőséget.*

1. Készítsen egy halmaz típust! Alkalmazzon osztályt! A halmazt rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket, egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (halmazbeolvasó operátor  $>>$ , halmazkiíró operátor  $<<$ ), definiáljon olyan barát-operátort is, amely kiszámítja két halmaz unióját! Az unió műveletigénye:  $O(m+n)$ , ahol  $m$  és  $n$  a két halmaz elemszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
2. Készítsen egy zsák típust! Alkalmazzon osztályt! A zsákokat rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket, egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (zsákbeolvasó operátor  $>>$ , zsákkiíró operátor  $<<$ ), definiáljon olyan barát-operátort is, amely kiszámítja két zsák unióját (a közös elemek előfordulása összegződik)! Az unió műveletigénye:  $O(m+n)$ , ahol  $m$  és  $n$  a két zsáknak megfelelő halmazok elemszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
3. Készítsen egy halmaz típust! Alkalmazzon osztályt! A halmazt rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket, egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (halmazbeolvasó operátor  $>>$ , halmazkiíró operátor  $<<$ ), definiáljon olyan barát-operátort is, amely kiszámítja két halmaz metszetét! A metszet műveletigénye:  $O(m+n)$ , ahol  $m$  és  $n$  a két halmaz elemszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
4. Készítsen egy zsák típust! Alkalmazzon osztályt! A zsákokat rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket, egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (zsákbeolvasó operátor  $>>$ , zsákkiíró operátor  $<<$ ), definiáljon olyan barát-operátort is, amely kiszámítja két zsák metszetét (a közös elemeket a kisebb előfordulási számmal)! A metszet műveletigénye:  $O(m+n)$ , ahol  $m$  és  $n$  a két zsáknak megfelelő halmazok elemszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
5. Készítsen egy sor típust! Alkalmazzon osztályt! A sorokat egyirányú láncolt listával ábrázolja! (A lista végére is mutasson egy pointer!) Implementálja a szokásos műveleteket és egy összefűző műveletet is, ami az első sorhoz sorban hozzáfűzi a második sor elemeit! (A második sor végül legyen üres!) Egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (sorbeolvasó operátor  $>>$ , sorkiíró operátor  $<<$ ), definiáljon olyan barát-operátort is, amely kiszámítja két sor összefűzését! Az összefűzés és a többi operáció műveletigénye is  $O(1)$ , kivéve a sorbeolvasó és sorkiíró operátorokat. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját! Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!

6. Készítsen egy sor típust! Alkalmazzon osztályt! A sorokat kétirányú fejelemes ciklikus láncolt listával ábrázolja! Implementálja a szokásos műveleteket, és egy összefűző műveletet is, ami az első sorhoz sorban hozzáfűzi a második sor elemeit! (A második sor végül legyen üres!) Egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (sorbeolvasó operátor $\gg$ , sorkiíró operátor $\ll$ ), definiáljon olyan barát-operátort is, amely kiszámítja két sor összefűzését! Az összefűzés és a többi operáció műveletigénye is  $O(1)$ , kivéve a sorbeolvasó és sorkiíró operátorokat. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
7. Készítsen egy zsák típust! Alkalmazzon osztályt! A zsákokat rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket, egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (beolvasó operátor $\gg$ , kiíró operátor $\ll$ ), definiáljon olyan barát-operátort is, amely kiszámítja két zsák különbségét (ha egy elem az első zsákban  $j$ -szer, a másodikban  $k$ -szor fordult elő, akkor a különbségben  $j-k$ -szor)! A különbségképzés műveletigénye:  $O(m+n)$ , ahol  $m$  és  $n$  a két zsáknak megfelelő halmazok elemszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
8. Készítsen egy halmaz típust! Alkalmazzon osztályt! A halmazt rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket, egészítse ki az osztályt a kényelmes és biztonságos használat érdekében megfelelő metódusokkal (beolvasó operátor $\gg$ , kiíró operátor $\ll$ ), definiáljon olyan barát-operátort is, amely kiszámítja két halmaz különbségét! A különbségképzés műveletigénye:  $O(m+n)$ , ahol  $m$  és  $n$  a két halmaz elemszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
9. Valósítsuk meg a nagyon nagyszámok típusát! Ábrázoljuk a számokat a számjegyeik láncolt listáival és implementáljuk az összeadást! Ne feledkezzünk meg a beolvasó és kiíró metódusokról sem! Az összeadás műveletigénye  $O(m+n)$ , a szorzásé  $O(m*n)$ , ahol  $m$  és  $n$  a két szám számjegyeinek száma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!
10. Valósítsuk meg a polinomok típusát! Ábrázoljuk a polinom együtthatóit egy láncolt listával és implementáljuk az összeadást! Ne feledkezzünk meg a beolvasó és kiíró metódusokról sem! Az összeadás műveletigénye  $O(m+n)$ , a szorzásé  $O(m*n)$ , ahol  $m$  és  $n$  a két polinom fokszáma. Alkalmazzon kivételkezelést, készítsen teszt környezetet, és bontsa modulokra a programját!