*Az alábbi feladatok típusait egy-egy **osztály segítségével** valósítsa meg. Az összes megvalósítandó típus összetett adatszerkezetű, azonos típusú elemek gyűjteménye, amelyet **láncolt listában** kell elhelyezni, ezért az osztályban meg kell valósítani a **másoló konstruktort** és az **értékadás operátort** is. Ahol a feladat szövege nem definiálja, az elemi típus az egész számok típusa. (Ne alkalmazzon sablonokat!)*

*Egy osztály szolgáltatásainak (metódusainak) bemutatásához olyan főprogramot kell készíteni, amelyik egy **menü segítségével** teszi lehetővé a metódusok tetszőleges sorrendben történő kipróbálását. A főprogram 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 is adjon lehetőséget.*

*Alkalmazzon **operátor felüldefiniálást** és **kivételkezelést**! Bontsa modulokra a programját!*

1. Készítsen egy halmaz típust! A halmazt rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket (elem betétele, kivétele, benne van-e egy adott elem, üres-e), egészítse ki az osztályt a halmaz tartalmát kiíró operátor<<-ral! Definiáljon olyan barát-operátorokat is, amely kiszámítja két halmaz unióját és különbségét! Az unió műveletigénye: $O(m+n)$, ahol m és n a két halmaz elemszáma.
2. Készítsen egy zsák típust! A zsákokat rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket (elem betétele, kivétele, benne van-e egy adott elem, hányszorosan van benne, üres-e), egészítse ki az osztályt a zsák tartalmát kiíró operátor<<-ral! Definiáljon olyan barát-operátorokat is, amely kiszámítja két zsák unióját (a közös elemek előfordulása összegződik) és különbségét (ha egy elem az első zsákban j-szer, a másodikban k-szor fordult elő, akkor a különbségben j-k-szor) ! Az unió és a különbség műveletigénye: $O(m+n)$, ahol m és n a két zsáknak megfelelő halmazok elemszáma.
3. Készítsen egy halmaz típust! A halmazt rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket (elem betétele, kivétele, benne van-e egy adott elem, üres-e), egészítse ki az osztályt a halmaz tartalmát kiíró operátor<<-ral! Definiáljon olyan barát-operátorokat is, amely kiszámítja két halmaz szimmetrikus differenciáját és metszetét! A metszet műveletigénye: $O(m+n)$, ahol m és n a két halmaz elemszáma.
4. Készítsen egy zsák típust! A zsákokat rendezett láncolt listával ábrázolja! Implementálja a szokásos műveleteket (elem betétele, kivétele, benne van-e egy adott elem, hányszorosan van benne, üres-e), egészítse ki az osztályt a zsák tartalmát kiíró operátor<<-ral! Definiáljon olyan barát-operátorokat is, amely kiszámítja két zsák szimmetrikus differenciáját (a közös elemek a multiplicitása az eredményben az eredeti multiplicitások különbségének abszolút értéke legyen) és metszetét (a közös elemeket a kisebb előfordulási számmal)! A szimmetrikus differencia és a metszet műveletigénye: $O(m+n)$, ahol m és n a két zsáknak megfelelő halmazok elemszáma.
5. Valósítsa meg a nagyon nagyszámok típusát! Ábrázoljuk a számokat a számjegyeik láncolt listáival és implementálja az összeadást és a szorzást operátor túldefiniálással! Ne feledkezzen meg a beolvasó (operátor>>) és kiíró (operátor<<) metódusokról sem! Az összeadás műveletigénye $O(m+n)$, a szorzásé $O(m*n)$, ahol m és n a két szám számjegyeinek száma.

6. Ábrázoljon egy négyzetes ritka mátrixot láncolt ábrázolással úgy, hogy a mátrix minden nem nulla elemét tartalmazó listaelem két láncolt listába is be legyen fűzve: a mátrix adott sorát és adott oszlopát reprezentáló listába! A sorokat reprezentáló listák az oszlopindexek szerint, az oszlopokat reprezentáló listák sorindex szerint rendezve tárolják a mátrix elemeket! Valósítsa meg a mátrix adott indexű elemének kiolvasását és felülírását elvégző műveletet a zárójel operátorral! Készítsen olyan műveletet, amely legfeljebb $n*(n-1)/2$ lépésben eldönti, hogy a mátrix diagonális-e! Írja meg a mátrixot kiíró operátor<<-t!

7. Valósítsa meg a polinomok típusát! Ábrázoljuk a polinom együtthatóit egy láncolt listával és implementálja az összeadást és a szorzást operátor túldefiniálással! Ne feledkezzen meg a beolvasó (operátor>>) és kiíró (operátor<<) metódusokról sem! Az összeadás műveletigénye $O(m+n)$, a szorzásé $O(m*n)$, ahol m és n a két polinom fokszáma!

8. Ábrázoljon egy ritka mátrixot láncolt ábrázolással úgy, hogy a mátrix nem nulla elemeit és az ahhoz tartozó sor és oszlop indexeket egy egyirányú láncolt listába fűzi a sor és oszlop indexpár sorfolytonos rendezettségnek megfelelően. Valósítsa meg a mátrix adott indexű elemének kiolvasását és felülírását elvégző műveletet a zárójel operátorral! Implementálja két ritka mátrix összeadását (operator+) valamint ritka mátrix ritka mátrixhoz való hozzáadását (operator+=) úgy, hogy a műveletigény legfeljebb a két mátrix elemszámának összege legyen! Írja meg a mátrixot kiíró operátor<<-t!

9. Valósítsa meg a ritka diagonális mátrixok típusát úgy, hogy a diagonális nem nulla elemeit egy láncolt listában helyezi el az indexükkel együtt! Valósítsa meg a mátrix adott indexű elemének kiolvasását és felülírását elvégző műveletet a zárójel operátorral! Implementálja két mátrix szorzását végrehajtó barát-operátorokat! Írja meg a mátrixot kiíró operátor<<-t!

10. Reprezentáljon egy sor típust egyirányú fejelemes listával (a lista végére is lehessen közvetlenül hivatkozni)! Implementálja a szokásos műveleteket (elem a végén be, az elején ki, üres-e, mi van az elején), továbbá egy összefűző műveletet (operator+) is, ami egy sorhoz hozzáfűzi egy másik sor elemeit (a művelet után a második sor legyen üres)! Egészítse ki egy a sor összes elemét kiíró operátor<<-ral. Az összefűzés és a többi operáció műveletigénye is $O(1)$, kivéve a sorkiíró operátort.