

Az alábbi feladatok megoldásához több olyan osztályt kell használni, amelyek egy közös ősosztályból származnak és felüldefiniálják az ősosztály virtuális metódusait.

1. Rögzítsen a síkon egy pontot, és töltsön fel egy láncolt listát különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Számolja meg, hogy a pontot hány síkidom tartalmazza! Minden síkidom reprezentálható a középpontjával és az oldalhosszal, illetve a sugárral, ha feltesszük, hogy a sokszögek esetében az egyik oldal párhuzamos a koordináta rendszer vízszintes tengelyével, és a többi csúcs ezen oldalra fektetett egyenes felett helyezkedik el. A síkidomokat szövegfájlból töltsse be! A fájl első sorában szerepeljen a síkidomok száma, majd ez egyes síkidomok. Az első jel azonosítja a síkidom fajtáját, amit követnek a középpont koordinátái és a szükséges hosszúság. A feladatokban a beolvasáson kívül a síkidomokat egységesen kezelje, ennek érdekében a síkidomokat leíró osztályokat egy közös ősosztályból származtassa!
2. Töltsön fel egy láncolt listát különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Határozza meg a legkisebb téglalapot, amely lefedi az összes síkidomot és oldalai párhuzamosak a tengelyekkel! Minden síkidom reprezentálható a középpontjával és az oldalhosszal, illetve a sugárral, ha feltesszük, hogy a sokszögek esetében az egyik oldal párhuzamos a koordináta rendszer vízszintes tengelyével, és a többi csúcs ezen oldalra fektetett egyenes felett helyezkedik el. A síkidomokat szövegfájlból töltsse be! A fájl első sorában szerepeljen a síkidomok száma, majd ez egyes síkidomok. Az első jel azonosítja a síkidom fajtáját, amit követnek a középpont koordinátái és a szükséges hosszúság. A feladatokban a beolvasáson kívül a síkidomokat egységesen kezelje, ennek érdekében a síkidomokat leíró osztályokat egy közös ősosztályból származtassa!
3. Rögzítsen a síkon egy pontot, és töltsön fel egy láncolt listát különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Keresse meg melyik síkidom van legközelebb a ponthoz! Közelségen kör esetén a körvonaltól vett távolságot, sokszög esetén a legközelebbi csúcstól mért távolságot értjük. Minden síkidom reprezentálható a középpontjával és az oldalhosszal, illetve a sugárral, ha feltesszük, hogy a sokszögek esetében az egyik oldal párhuzamos a koordináta rendszer vízszintes tengelyével, és a többi csúcs ezen oldalra fektetett egyenes felett helyezkedik el. A síkidomokat szövegfájlból töltsse be! A fájl első sorában szerepeljen a síkidomok száma, majd ez egyes síkidomok. Az első jel azonosítja a síkidom fajtáját, amit követnek a középpont koordinátái és a szükséges hosszúság. A feladatokban a beolvasáson kívül a síkidomokat egységesen kezelje, ennek érdekében a síkidomokat leíró osztályokat egy közös ősosztályból származtassa!
4. Töltsön fel egy láncolt listát különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Adja meg azt a síkidomot, amelynek a területe és a kerülete a legkisebb mértékben tér el! Minden síkidom reprezentálható a középpontjával és az oldalhosszal, illetve a sugárral, ha feltesszük, hogy a sokszögek esetében az egyik oldal párhuzamos a koordináta rendszer vízszintes tengelyével, és a többi csúcs ezen oldalra fektetett egyenes felett helyezkedik el. A síkidomokat szövegfájlból töltsse be! A fájl első sorában szerepeljen a síkidomok száma, majd ez egyes síkidomok. Az első jel azonosítja a síkidom fajtáját, amit követnek a középpont koordinátái és a szükséges hosszúság. A feladatokban a beolvasáson kívül a síkidomokat egységesen kezelje, ennek érdekében a síkidomokat leíró osztályokat egy közös ősosztályból származtassa!
5. Töltsön fel egy láncolt listát különféle szabályos (kör, szabályos háromszög, négyzet, szabályos hatszög) síkidomokkal! Adja meg melyik síkidom befoglaló téglalapja a legnagyobb területű! Egy síkidom befoglaló téglalapja lefedi a síkidomot, oldalai

párhuzamosak a tengelyekkel. Minden síkidom reprezentálható a középpontjával és az oldalhosszal, illetve a sugárral, ha feltesszük, hogy a sokszögek esetében az egyik oldal párhuzamos a koordináta rendszer vízszintes tengelyével, és a többi csúcs ezen oldalra fektetett egyenes felett helyezkedik el. A síkidomokat szövegfájlból töltse be! A fájl első sorában szerepeljen a síkidomok száma, majd ez egyes síkidomok. Az első jel azonosítja a síkidom fajtáját, amit követnek a középpont koordinátái és a szükséges hosszúság. A feladatokban a beolvasáson kívül a síkidomokat egységesen kezelje, ennek érdekében a síkidomokat leíró osztályokat egy közös őszosztályból származtassa!

6. Egy szöveges állomány háromféle síkidom adatait (négyzetnél az alapot, téglalpnál az alapot és az oldalt, rombusznál az alapot és az alapok által bezárt szöget) tartalmazza. Minden sorban egy-egy síkidom adatai találhatóak. A sor első számjegye a síkidom fajtájára utal (0, ha négyzet; 1, ha téglalap; 2, ha rombusz), ezután szóközökkel elválasztva a síkidom fajtájától függően egy vagy két valós szám. Definiálja külön-külön az egyes síkidomok-típusok osztályait úgy, hogy a négyzetét az alább megadott absztrakt osztályból származtatja, a másik kettőt pedig a négyzet osztályából!

```
class Absztrakt
{
    public:
        virtual ~Absztrakt() {}
        virtual std::string Nev() = 0;
        virtual double Terulet() { return Alap()*Magassag(); }
        virtual double Kerulet() { return 2*(Alap()+Oldal()); }
        virtual double Alap() = 0;
        virtual double Oldal() { return Alap(); }
        virtual double Magassag() { return Oldal(); }
};
```

Készítsen olyan programot, amely a szöveges állomány alapján létrehozza a megfelelő síkidom-objektumokat, ezeket területük szerint is, és kerületük szerint is növekedő sorrendben tárolja! Használjon ehhez láncolt listát! NEM KÉT KÜLÖNBÖZŐ LÁNCOLT LISTÁT KELL LÉTREHOZNI, HANEM EGY KÉTSZERESEN LÁNCOLTAT! Ez pontosan annyi listaelemet tartalmaz, ahány síkidomunk van, de egy listaelemnek két mutatója is van: az egyik a terület szerinti sorrendet, a másik a kerület szerinti sorrendet mutatja. Ennek a listának egymás után kétszer történő bejárásával írja ki a szabványos kimenetre terület szerint növekedő sorrendben a síkidomok nevét és területét, majd kerület szerint növekedő sorrendben a síkidomok nevét és kerületét! Végül bontsa le a listát!

7. Egy szöveges állomány háromféle síkidom adatait (négyzetnél az alapot, téglalpnál az alapot és az oldalt, rombusznál az alapot és az alapok által bezárt szöget) tartalmazza. Minden sorban egy-egy síkidom adatai találhatóak. A sor első számjegye a síkidom fajtájára utal (0, ha négyzet; 1, ha téglalap; 2, ha rombusz), ezután szóközökkel elválasztva a síkidom fajtájától függően egy vagy két valós szám. Definiálja külön-külön az egyes síkidomok-típusok osztályait úgy, hogy mindet az alább megadott absztrakt osztályból származtatja!

```
class Absztrakt
{
    protected:
        double alap;
    public:
        virtual ~Absztrakt() {}
        virtual std::string Nev() = 0;
        virtual double Terulet() { return alap*Magassag(); }
        virtual double Kerulet() { return 2*(alap+Oldal()); }
};
```

```

    virtual double Oldal()    { return alap;}
    virtual double Magassag(){ return Oldal();}
};

```

Készítsen olyan programot, amely a szöveges állomány alapján létrehozza a megfelelő síkidom-objektumokat, ezeket területük szerint növekedő sorrendben befűzi egy KÉTIRÁNYÚ LÁNCOLT LISTÁBA! A láncolt lista előre haladó, majd visszafelé haladó bejárásával írja ki a szabványos kimenetre a terület szerint növekedő, majd terület szerint csökkenő sorrendben is a síkidomok nevét és területét, végül bontsa le a listát!

8. Egy szöveges állomány egy lakás helységeinek adatait (lakószobáknál az oldalhosszakat és az ablakok számát, egyéb helységek esetén az alapterületet és az ablakok számát) tartalmazza. Minden sorban egy-egy helység adatai találhatók. A sor első számjegye a helység fajtájára utal (1, ha szoba; 2, ha egyéb helység), ezután szóközzel elválasztva a helység fajtájától függően egy vagy két valós szám, majd az ablakok számát megadó egész. Definiálja külön-külön az egyes helység-típusok osztályait! Készítsen olyan programot, amely a szöveges állomány alapján létrehozza a megfelelő helység-objektumokat, amelyeket ablakszám szerint csökkenő sorrendben befűzi egy láncolt listába. Ezután, mialatt lebontja a láncolt listát, írja ki a szabványos kimenetre az eltárolt helységek fajtáját, alapterületét, és ablakszámát.

9. Készítsünk C++ programot a következő feladat megoldására!

Egy többnapos versenyen lények vesznek részt. A versenyt az a lény nyeri, aki életben marad, és a legnagyobb távolságot teszi meg. Kezdetben minden lény valamennyi vízzel rendelkezik, és a megtett távolság 0. A verseny során háromféle nap lehetséges: napos, felhős és esős. Ezekre a különböző fajtájú lények eltérő módon reagálnak vízfogyasztás és haladás szempontjából. Minden lény először a rendelkezésére álló víz mennyiségét változtatja meg, ezután ha tud, mozog. Bármely lény elpusztul, ha a vize elfogy (0 lesz az érték), ezután értelemszerűen semmilyen tevékenységre sem képes.

Minden lény jellemzői: az egyedi neve (string), a rendelkezésre álló víz mennyisége (egész), a maximálisan tárolható víz mennyisége (egész), hogy él-e (logikai), illetve az eddig megtett távolság (egész). A versenyen részt vevő lények fajtái a következők: homokjáró, szivacs, lépegető.

A következő táblázat tartalmazza az egyes fajták jellemzőit.

fajta	víz változtatás			távolság			max.víz
	napos	felhős	esős	napos	felhős	esős	
homokjáró	-1	0	3	3	1	0	8
szivacs	-4	-1	6	0	1	3	20
lépegető	-2	-1	3	1	2	1	12

Az egyes lények a vízkészlet megváltoztatása során nem léphetik túl a fajtára jellemző maximális értéket, legfeljebb azt érhetik el.

A program egy szövegfájlból olvassa be a verseny adatait! Az első sorban az induló lények száma szerepel. A következő sorok tartalmazzák a lények adatait szóközzel elválasztva: a lény nevét, a fajtáját és a kezdetben rendelkezésére álló víz mennyiségét. A fajtát egy karakter azonosítja: h - homokjáró, s - szivacs, l - lépegető.

A lényeket leíró részt követő sorban a verseny napjai szerepelnek egy karaktorsorozatban. Az egyes jelek értelmezése: n - napos, f - felhős, e - esős.

A program kérje be a fájl nevét, majd jelenítse meg a nyertes nevét! (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

```
4
Vandor h 4
Seta l 7
Csuszo s 12
Siklo s 10
nffeeennf
```

(Javaslat: Hozza létre az absztrakt lény osztályt, amelyből származtatja a homokjáró, szivacs és lépegető osztályokat. A konstruktor paramétere legyen a név és a kezdeti víz mennyisége. Vezessen be három műveletet a napoknak (napos, felhős, esős), amelyek a vízfogyasztást szabályozzák az élet vizsgálatával együtt, illetve mozgatják a lényt. A végeredmény meghatározásához kell még 3 művelet: él-e a lény, a név illetve a megtett távolság lekérdezése.)

10. Készítsünk C++ programot a következő feladat megoldására!

Egy bolygón különböző fajtájú növények élnek, minden növény tápanyagot használ. Ha egy növény tápanyaga elfogy (a mennyisége 0 lesz), a növény elpusztul. A bolygón három fajta sugárzást különböztetünk meg: alfa sugárzás, delta sugárzás, nincs sugárzás. A sugárzásra a különböző fajtájú élő növények eltérő módon reagálnak. A reakció tartalmazza a tápanyag változását, illetve a következő napi sugárzás befolyásolását. A másnapi sugárzás alakulása: ha az alfa sugárzásra beérkezett igények összege legalább hárommal meghaladja a delta sugárzás igényeinek összegét, akkor alfa sugárzás lesz; ha a delta sugárzásra igaz ugyanez, akkor delta sugárzás lesz; ha a két igény közti eltérés háromnál kisebb, akkor nincs sugárzás. Az első nap sugárzás nélküli.

Minden növény jellemzői: az egyedi neve (string), a rendelkezésre álló tápanyag mennyisége (egész), hogy él-e (logikai). A szimulációban részt vevő növények fajtái a következők: puffancs, deltafa, parabokor. A következőkben megadjuk, hogy az egyes fajták miként reagálnak a különböző sugárzásokra. Először a tápanyag változik, és ha a növény ezután él, akkor befolyásolhatja a sugárzást.

Puffancs: Alfa sugárzás hatására a tápanyag mennyisége kettővel nő, sugárzás mentes napon a tápanyag eggyel csökken, delta sugárzás esetén a tápanyag kettővel csökken. Minden esetben úgy befolyásolja a másnapi sugárzást, hogy 10 - tápanyag értékben növeli az alfa sugárzás bekövetkezését. Ez a fajta akkor is elpusztul, ha a tápanyag mennyisége 10 fölé emelkedik.

Deltafa: Alfa sugárzás hatására a tápanyag mennyisége hárommal csökken, sugárzás nélküli napon a tápanyag eggyel csökken, delta sugárzás hatására a tápanyag négyvel nő. Ha a tápanyag mennyisége 5-nél kisebb, akkor 4 értékben növeli a delta sugárzás bekövetkezését, ha 5 és 10 közé esik, akkor 1 értékben növeli a delta sugárzás bekövetkezését, ha 10-nél több, akkor nem befolyásolja a másnapi sugárzást.

Parabokor: Akár alfa, akár delta sugárzás hatására a tápanyag mennyisége eggyel nő. Sugárzás nélküli napon a tápanyag eggyel csökken. A másnapi sugárzást nem befolyásolja.

A program a növények viselkedését és a sugárzást szimulálja, egy szövegfájlból olvassa be a szimuláció adatait! Az első sorban a növények száma szerepel. A következő sorok tartalmazzák a növények adatait szóközzel elválasztva: a növény nevét, a fajtáját és a kezdetben rendelkezésére álló tápanyag mennyiségét. A fajtát egy karakter azonosítja: a - puffancs, d - deltafa, p - parabokor. A növényeket leíró részt követő sorban a szimuláció napjainak száma adott egész számként. A program kérje be a fájl nevét, majd jelenítse meg a túlélők nevét! (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

```
4
Falánk a 7
Sudár d 5
Köpcös p 4
Nyúlánk d 3
10
```

(Javaslat: Hozza létre az absztrakt növény osztályt, amelyből az egyes fajták osztályait. A konstruktor paramétere legyen a név és a tápanyag kezdeti mennyisége. Egy nap elteltére is vezessen be művelet(ek)et, a visszatérő érték adja meg, hogy a növény milyen módon befolyásolja a következő nap sugárzását. A befolyásolás lehet egy egész szám, ami pozitív alfa sugárzás esetén, negatív delta sugárzás esetén, nulla, ha a növény nem befolyásolja a

másnap sugárzást. A végeredmény meghatározásához kell még 2 művelet: él-e a lény, illetve a név lekérdezése.)