

Feladat: Mátrixok beolvasása fájlból.

Legyen egy **Matrix** nevű template-s osztály, ami megkapja, hogy a mátrixba milyen típusú elemek lesznek.

private adatai: egy **név**, (amelyet a beolvas metódus fog majd feltölteni) és egy olyan láncolt **lista**, amely map-eket (<adattípus, int>)(int helyett bármi jó) tartalmaz (ezek az egyes sorok tartalmát tárolják, rendezett formában)

+beolvas publikus metódus, amely paraméterként kapja a betölteni kívánt fájl nevét. Nyissa meg a fájlt, ha nem lehet ezt jelezze (cerr-en), majd olvasson be egy-egy sort és ezeket dolgozza fel, egy map-be, amit felvisz a listára (evvel új sorral bővítve a mátrixot). *Ajánlott egy sor feldolgozó private metódus erre a célra.*

-teljes private metódus amely logikai igazsággal tér vissza, ha a matrix sorai, egyforma hosszúak.

-kiir private metódus ami egy ostream kap paraméterül (pl: cout vagy cerr), és erre kiírja, a nevét és a tartalmát.

<< operátor a kiíráshoz, ha mátrix nem teljes akkor a kivételkezeléssel a cerr-re hibaüzenettel egyébként meg a cout-ra írja ki a mátrixot, a kiir metódus segítségével.

Legyen egy template-s **compare** függvény amely, map-et kap paraméterül, megnézi, hogy mennyi az adott map-ben lévő elemek összege, mennyi és igazsággal tér vissza, ha az első összértéke kisebb, mint a másodiké.

+sort(rendez) publikus metódus amely egy függvényt vár paraméterül

pl: void sort(bool fg(const map<T, int>& a,const map<T, int>& b))

és meghívja evvel a **lista adattag sort** metódusát, amely e függvény logikai értéke alapján felcserél két elemet a listán, amíg rendezetté nem válik.

```
int main() {
    Matrix<int> matrix;
    matrix.beolvas("1.txt");
    matrix.sort(compare<int>);
    cout << matrix;

    Matrix<int> matrix2;
    matrix2.beolvas("2.txt");
    matrix2.sort(compare<int>);
    cout << matrix2;

    Matrix<char> matrix3;
    matrix3.beolvas("3.txt");
    matrix3.sort(compare<char>);
    cout << matrix3;

    return 0;
}
```

1.txt:
0 1 2 3
1 2 3 4
2 3 4 5
3 4 5 6
4 5 6 7

Nem teljes 2.txt:
1 2 4 5
0 2 3 7
1 4 5 6
8 9

3.txt:
a b c d e f
b c d e f g