FEJLETT PROGRAMOZÁSI NYELVEK, 2009
13. GYAKORLAT – Adatfolyamok

Cél:

- Adatfolyamok, fájlfolyamok, adatfolyam-iterátorok
  - `istream, ostream,`
  - `fstream, ifstream, ofstream`
  - `istream_iterator, ostream_iterator`
- Manipulátorok
  - `noskipws`

1. Adatfolyam nagybetűsítése – `noskipws` manipulátor

Készítsen programot, amely adatfolyamok segítségével nagybetűsít egy állományt. A nagybetűsítést végezze a `transform` algoritmus segítségével és *adatfolyam iterátorokkal.*

2. Indexelhető bemeneti állomány, `ifstream` bővítése

Készítsünk egy olyan osztályt (`myifstream`), amelyben az index operátor segítségével közvetlenül olvashatjuk be a fájl karaktereit. Az osztályt úgy kell elkészíteni, hogy helyesen működjön a következő programrészletre:

```
myifstream f("be.txt");
for( int i=0; i<f.filesize(); ++i )
      cout<<f[i]<<" ";
cout<<endl;
```

3. Indexelhető bemeneti/kimeneti állomány, `fstream` bővítése

Általánosítsa az előző feladatot. Készítse el az index operátort úgy az adatfolyam osztályhoz (`IndexFileStream`), hogy írásra és olvasásra is használható legyen. Az osztályt úgy kell elkészíteni, hogy helyesen működjön a következő nagybetűsítést végző programrészletre:

```
IndexFileStream f("be.txt");
for( int i=0; i<f.filesize(); ++i )
            cout<<f[i]<<" ";
cout<<endl;
for( int i=0; i<f.filesize(); ++i )
            f[i]=::toupper( f[i] );
cout<<endl;
for( int i=0; i<f.filesize(); ++i )
            cout<<f[i]<<" ";
cout<<endl;
```

Ötlet: az index operátor adjon vissza egy olyan objektumot (proxy), amely egy fájlpozíciót azonosít. Az ezen objektumra vonatkozó értékadás írja fájlba a megfelelő adatokat, míg az objektumnak char típusra való automatikus konverziója jelentse a megfelelő karakter beolvasását az állományból.