

Exercise 3 (10 points) – can be done in pair or individually

- The first lines of all source files must be comments containing names & IDs of all members. Also create file readme.txt containing names & IDs of all members
- Put all files (source, input, readme.txt) in folder **Ex3_xxx** where **xxx = ID of the group representative**. That is, your source files must be in package Ex3_xxx and input files must be read from this path
- The group representative zips Ex3_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted

Keep your source code for Exercise 4

1. Copy class `Series` to your source file. Complete this class to make it concrete, but do not change the visibility of each member

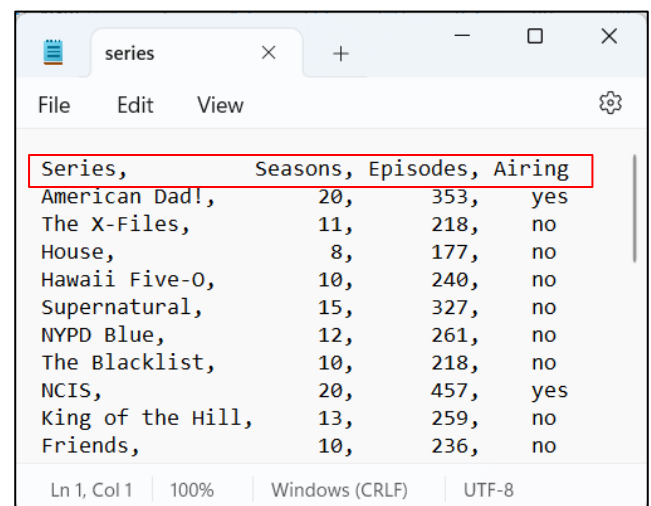
```
class Series implements Comparable<Series> {
    private String name;
    private int seasons, episodes;
    private String airing;

    public int compareTo(Series other) {
        // add your code according to conditions in 2.2-2.5
    }
}
```

2. Write another class that acts as the main class

- 2.1 Create an `ArrayList` of `Series`. Read each line of input file into each `Series` object and add this object to the `ArrayList`
 - Col 0 : series
 - Col 1 : total seasons
 - Col 2 : total episodes
 - Col 3 : whether it is still airing

The first line contains column names. **Don't remove this line from input file**, but make your program skip it.



Series,	Seasons,	Episodes,	Airing
American Dad!,	20,	353,	yes
The X-Files,	11,	218,	no
House,	8,	177,	no
Hawaii Five-0,	10,	240,	no
Supernatural,	15,	327,	no
NYPD Blue,	12,	261,	no
The Blacklist,	10,	218,	no
NCIS,	20,	457,	yes
King of the Hill,	13,	259,	no
Friends,	10,	236,	no

- 2.2 Sort `Series` objects in decreasing order of seasons
- 2.3 If seasons are equal, place active series (still airing) before non-active series
- 2.4 Among active series with the same number of seasons, sort them in decreasing order of episodes, then by name (if episodes are equal)
- 2.5 Among non-active series with the same number of seasons, sort them in decreasing order of episodes, then by name (if episodes are equal)

3. Print the sorting results in proper format

Note - Class String in Java already implements compareTo and compareToIgnoreCase

```
String s1 = "Friends";  
String s2 = "Frasier";  
int x = s1.compareToIgnoreCase(s2);    // x = 8, i.e. Friends comes later (i - a = 8)  
int x = s2.compareToIgnoreCase(s1);    // x = -8, i.e. Frasier comes first (a - i = -8)
```

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ solu
```

Series	Seasons	Airing	Episodes
=====			
The Simpsons	34	yes	750
NCIS	20	yes	457
American Dad!	20	yes	353
Gunsmoke	20	no	635
Grey's Anatomy	19	yes	420
Lassie	19	no	591
ER	15	no	331
Supernatural	15	no	327
Bob's Burger	13	yes	260
King of the Hill	13	no	259
Two and a Half Men	12	no	262
NYPD Blue	12	no	261
Bones	12	no	246
Chicago Fire	11	yes	239
Frasier	11	no	264
Will & Grace	11	no	246
The X-Files	11	no	218
Chicago P.D.	10	yes	208
Hawaii Five-O	10	no	240
Friends	10	no	236
Smallville	10	no	218
The Blacklist	10	no	218
Chicago Med	8	yes	163
Futurama	8	yes	142
House	8	no	177
Arrow	8	no	170
Barney Miller	8	no	170

Equal seasons → active before non-active

- Among active series → sort by episodes, then by name
- Among non-active series → sort by episodes, then by name