Exercise 9 (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 Ex9_xxx where xxx = ID of the group representative. That is, your source files must be in package Ex9_xxx and input files must be read from this path

- The group representative zips Ex9_xxx & submits it to Google Classroom. The other members submit only readme.txt. Email submission is not accepted

===

1. Copy class Series to your source file. The class is already fully implemented. <u>It must not be changed</u>. Take note on the compactness of class in functional programming style

```java
class Series implements Comparable<Series> {
  private String   name;
  private int      seasons, episodes;
  private String   airing;
  private double   result;          // dummy variable to keep calculation result

  public  Series(String na, int se, int ep, String air) {
    name     = na;
    seasons  = se;
    episodes = ep;
    airing   = air;
  }

  public void setName(String na)     { name = na; }
  public void setSeasons(int se)     { seasons = se; }
  public void setEpisodes(int ep)    { episodes = ep; }
  public void setAiring(String air)  { airing = air; }
  public void setResult(double res)  { result = res; }

  public String  getName()           { return name; }
  public int     getSeasons()        { return seasons; }
  public int     getEpisodes()       { return episodes; }
  public String  getAiring()         { return airing; }
  public double  getResult()         { return result; }

  public int compareTo(Series other) {
    int x = Double.compare(other.result, result);          // decreasing order
    if (x != 0) return x;
    else        return name.compareToIgnoreCase(other.name);   // alphabetical order
  }
}
```

2. Create <u>ArrayList</u> of Series. Read each line of series.txt into Series object & add it to the ArrayList. Each line consists of: name, seasons, episodes, airing
** You can reuse your code from Exercises 3-4

3. Get Streams of Series from ArrayList in (2). Use lambda and stream processing (such as filter, map, forEach, sorted, reduce) to get the following output. Loops are not allowed

3.1 Print only stop-airing series (name and seasons), sorted in increasing order of season then by alphabetical order of name. Series names must be in all lowercase & left aligned

3.2 Read episode threshold from user. Print all series whose episodes >= threshold (name, episodes, episodes per season), sorted in decreasing order of episodes/season then by alphabetical order of name. Country names must be in all uppercase & right aligned. Also report the number of such series and avg episodes/season among them

3.3 Using the same threshold as in (3.2), print still-airing series whose episodes < threshold (name and episodes), sorted in decreasing order of episodes then by alphabetical order of name. Country names must be in all uppercase & right aligned. Also report the number of such series and total episodes among them

Note : Use negate for reverse condition
```
    Predicate<Series> seasonPredicateLambda = (Series sr) -> sr.getSeasons() > 15;
    mystream.filter( seasonPredicateLambda );               // seasons > 15
    mystream.filter( seasonPredicateLambda.negate() );      // seasons <= 15
```

```
=== Stop-airing series, sorted by seasons ===
arrow                   seasons =   8
barney miller           seasons =   8
house                   seasons =   8
friends                 seasons =  10
hawaii five-o           seasons =  10
smallville              seasons =  10
the blacklist           seasons =  10
frasier                 seasons =  11
the x-files             seasons =  11
will & grace            seasons =  11
bones                   seasons =  12
nypd blue               seasons =  12
two and a half men      seasons =  12
king of the hill        seasons =  13
er                      seasons =  15
supernatural            seasons =  15
lassie                  seasons =  19
gunsmoke                seasons =  20

Enter min episodes =
400

=== All series with >= 400 episodes, sorted by episodes per season ===
        GUNSMOKE    episodes = 635     episodes per season = 31.75
          LASSIE    episodes = 591     episodes per season = 31.11
            NCIS    episodes = 457     episodes per season = 22.85
   GREY'S ANATOMY   episodes = 420     episodes per season = 22.11
     THE SIMPSONS   episodes = 750     episodes per season = 22.06

Number of series      = 5
Avg. episodes per season = 25.97


=== Still-airing series with < 400 episodes, sorted by episodes ===
     AMERICAN DAD!   episodes = 353
      BOB'S BURGER   episodes = 260
     CHICAGO FIRE    episodes = 239
     CHICAGO P.D.    episodes = 208
      CHICAGO MED    episodes = 163
         FUTURAMA    episodes = 142

Number of series = 6
Total episodes   = 1,365

-----------------------------------------------------------------------
```