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

==========

1. Complete class Language. Add variables and methods as needed but the given variables must remain private.

```
class Language {
  private String name;
  private int    releaseYear;
  private double stackQuestions;      // Stackoverflow questions in millions
  private double githubRepos;         // Github repositories     in millions
  private double githubTopStars;      // Github top-repo stars   in thousands
}
```

2. Create ArrayList of Language. Read each line of input file into Language object & add the object to ArrayList. First line consists of variable names. Each of remaining lines consists of: name, release year, Stackoverflow questions (millions), Github repositories (millions), Github top-repository stars (thousands)

3. Ask user to choose sorting criterion and print results. Output columns are in the same order as input columns

3.1 **n,N** sort Languages by alphabetical order of name

3.2 **y,Y** sort Languages in increasing order of year & decreasing order of next columns (Stackoverflow questions, Github repos, Github stars). But if year & all next columns are equal, sort them by alphabetical order of name

3.3 **q,Q** sort Languages in decreasing order of Stackoverflow questions & increasing order of next columns (Github repos, Github stars). But if Stackoverflow questions & all next columns are equal, sort them by alphabetical order of name

3.4 **r,R** sort Languages in decreasing order of Github repos & increasing order of next column (Github stars). But if Github repos & all next columns are equal, sort them by alphabetical order of name

3.5 **s,S** sort Languages in decreasing order of Github stars, then by alphabetical order of name

Implement Comparator classes for sorting

```
class SortLanguageByName           implements Comparator<Language>  {...}
class SortLanguageByYear           implements Comparator<Language>  {...}
class SortLanguageByStackoverflow  implements Comparator<Language>  {...}
class SortLanguageByGithubRepos    implements Comparator<Language>  {...}
class SortLanguageByGithubStars    implements Comparator<Language>  {...}
```

4. Program must be able to loop for new sorting criteria

Note : When diff between 2 double values is <1 (e.g. between 1.5 and 1.8),
use Double.compare(a1.getValue(), a2.getValue())
instead of (int)(a1.getValue() – a2.getValue())

```
Sort by >> n = name
           y = year
           q = Stackoverflow questions
           r = Github repos
           s = Github stars
      others = quit
n
```

Language	Release Year	Stackoverflow Questions(M)	Github Repos(M)	GitHub Top Stars(K)
C	1972	0.5	1.5	1.1
C#	2000	1.6	1.7	0.7
C++	1983	1.0	1.2	0.7
Go	2009	0.1	0.1	6.6
Groovy	2004	0.1	0.1	0.8
HTML	1993	1.2	4.4	0.9
Java	1995	1.8	5.0	1.6
JavaScript	1995	2.5	6.2	1.7
Kotlin	2011	0.1	0.1	0.6
Lua	1993	0.1	0.1	0.5
Objective-C	1984	0.3	0.4	1.2
PHP	1995	1.8	1.7	1.1
Python	1991	2.1	3.1	1.1
R	1993	0.5	0.4	0.4
Ruby	1995	0.2	1.5	1.5
Scala	2003	0.2	0.1	1.2
Swift	2014	0.3	0.5	1.6
XML	1996	0.2	0.1	0.5

```
Sort by >> n = name
           y = year
           q = Stackoverflow questions
           r = Github repos
           s = Github stars
      others = quit
y
```

Language	Release Year	Stackoverflow Questions(M)	Github Repos(M)	GitHub Top Stars(K)
C	1972	0.5	1.5	1.1
C++	1983	1.0	1.2	0.7
Objective-C	1984	0.3	0.4	1.2
Python	1991	2.1	3.1	1.1
HTML	1993	1.2	4.4	0.9
R	1993	0.5	0.4	0.4
Lua	1993	0.1	0.1	0.5
JavaScript	1995	2.5	6.2	1.7
Java	1995	1.8	5.0	1.6
PHP	1995	1.8	1.7	1.1
Ruby	1995	0.2	1.5	1.5
XML	1996	0.2	0.1	0.5
C#	2000	1.6	1.7	0.7
Scala	2003	0.2	0.1	1.2
Groovy	2004	0.1	0.1	0.8
Go	2009	0.1	0.1	6.6
Kotlin	2011	0.1	0.1	0.6
Swift	2014	0.3	0.5	1.6

```
Sort by >> n = name
          y = year
          q = Stackoverflow questions
          r = Github repos
          s = Github stars
     others = quit
q
```

Language	Release Year	Stackoverflow Questions(M)	Github Repos(M)	GitHub Top Stars(K)
JavaScript	1995	2.5	6.2	1.7
Python	1991	2.1	3.1	1.1
PHP	1995	1.8	1.7	1.1
Java	1995	1.8	5.0	1.6
C#	2000	1.6	1.7	0.7
HTML	1993	1.2	4.4	0.9
C++	1983	1.0	1.2	0.7
R	1993	0.5	0.4	0.4
C	1972	0.5	1.5	1.1
Objective-C	1984	0.3	0.4	1.2
Swift	2014	0.3	0.5	1.6
XML	1996	0.2	0.1	0.5
Scala	2003	0.2	0.1	1.2
Ruby	1995	0.2	1.5	1.5
Lua	1993	0.1	0.1	0.5
Kotlin	2011	0.1	0.1	0.6
Groovy	2004	0.1	0.1	0.8
Go	2009	0.1	0.1	6.6

```
Sort by >> n = name
          y = year
          q = Stackoverflow questions
          r = Github repos
          s = Github stars
     others = quit
r
```

Language	Release Year	Stackoverflow Questions(M)	Github Repos(M)	GitHub Top Stars(K)
JavaScript	1995	2.5	6.2	1.7
Java	1995	1.8	5.0	1.6
HTML	1993	1.2	4.4	0.9
Python	1991	2.1	3.1	1.1
C#	2000	1.6	1.7	0.7
PHP	1995	1.8	1.7	1.1
C	1972	0.5	1.5	1.1
Ruby	1995	0.2	1.5	1.5
C++	1983	1.0	1.2	0.7
Swift	2014	0.3	0.5	1.6
R	1993	0.5	0.4	0.4
Objective-C	1984	0.3	0.4	1.2
Lua	1993	0.1	0.1	0.5
XML	1996	0.2	0.1	0.5
Kotlin	2011	0.1	0.1	0.6
Groovy	2004	0.1	0.1	0.8
Scala	2003	0.2	0.1	1.2
Go	2009	0.1	0.1	6.6

```
Sort by >> n = name
          y = year
          q = Stackoverflow questions
          r = Github repos
          s = Github stars
     others = quit
s
```

Language	Release Year	Stackoverflow Questions(M)	Github Repos(M)	GitHub Top Stars(K)
Go	2009	0.1	0.1	6.6
JavaScript	1995	2.5	6.2	1.7
Java	1995	1.8	5.0	1.6
Swift	2014	0.3	0.5	1.6
Ruby	1995	0.2	1.5	1.5
Objective-C	1984	0.3	0.4	1.2
Scala	2003	0.2	0.1	1.2
C	1972	0.5	1.5	1.1
PHP	1995	1.8	1.7	1.1
Python	1991	2.1	3.1	1.1
HTML	1993	1.2	4.4	0.9
Groovy	2004	0.1	0.1	0.8
C#	2000	1.6	1.7	0.7
C++	1983	1.0	1.2	0.7
Kotlin	2011	0.1	0.1	0.6
Lua	1993	0.1	0.1	0.5
XML	1996	0.2	0.1	0.5
R	1993	0.5	0.4	0.4