Concurrent and Parallel Programming in Java 8

part III – Basic streams

1. Given the following definitions:

```
Book b1 = new Book("Java 8 lambdas", "Richard Warbuton", 182, "O'Reilly");
Book b2 = new Book("Java 8 in action", "Raoul-Gabriel Urma", 497, "Manning");
Book b3 = new Book("Functional thinking", "Neal Ford", 179, "O'Reilly");
Book b4 = new Book("Learning scala", "Jason Swartz", 255, "O'Reilly");
Book b5 = new Book("Parallel and concurrent programming in Haskell",
                   "Simon Marlow", 321, "O'Reilly");
Book b6 = new Book("Presentation patterns", "Neal Ford", 265,
                   "Addisson Wesley");
List<Book> books = Arrays.asList(b1, b2, b3, b4, b5, b6);
```

Use streams operations to obtain:

a. a list with the number of pages of all books.
b. a list with all the author names without duplicates.
c. a list with all book titles from O'Reilly.
d. a list with all the titles of the books about Java.
e. an ordered list with all surnames of the authors without duplicates
f. the book with the longest title.
g. the title of the book with the largest number of pages.
h. the average number of pages of all books published by O'Reilly.
i. the number of books published by each publisher(as a map with publisher as index).
j. the average number of pages of the books published by each author (as a map with author as index)

2. Define an infinite stream, which can generate: the powers of two as an ordered sequence: 2, 4, 8, 16, 32, etc.

3. Define the method `int powerOf2(int n)`, which returns the smallest power of two which is not smaller than *n*, by using the stream defined in the previous exercise.

4. Define an infinite stream, which can generate the sequence of arrays with two equal integer values: [0,0], [1,1], [2,2], etc.