Lab Report (Advanced Stream API methods)

I/we the undersigned, promise that the submitted lab report is/are my/our own work. While I/we was/were
free to discuss ideas with others, the work contained is my/our own. I/we recognize that should this not be the
case; I/we will be subject to penalties as outlined in the course syllabus.
(By typing in your name below, you agree to Academic Integrity and honesty)

Name: **Nero Hamidi**
Red-Id: **827723033**

Reflection:
In a short paragraph explain your learning in this lab and how stream API helps in data processing.
Explain
- flatmap
- anyMatch
- average
- sum

In this lab, we went over stream API and how it helps data processing. According to an article on stream API by Oracle (the current owner of Java), “streams can leverage multi-core architectures without you having to write a single line of multithread code.” Essentially, stream API makes data processing more efficient. Another useful thing about stream API is how it makes your code more readable, allowing the programmer to manipulate data easier. One method in stream API is `flatMap`, which flattens a stream of collections into a single stream. This makes the data simpler to use and further manipulate. We also learned about `anyMatch`, which checks if any element in the stream matches a set condition. Also, we learned about

`average`, which calculates the average value for a stream of numbers. Lastly, we used `sum` to calculate the sum of a stream of numbers. Many of the names of these methods describe the action they perform, which makes code more readable and easier to understand. Additionally, these are some of the most common actions performed on sets of data, so having a simple method call for them makes dealing with data much more efficient.