1. The Fibonacci sequence is defined by

$$F(n) = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F(n-1) + F(n-2) & n > 1 \end{cases}$$

Write an efficient function that returns the index i for the first Fibonacci number that has more than 1000 digits i.e. F(i) has 1000 or more decimal digits and F(i-1) has less than 1000 digits. Note that the number 1000 only has 4 digits.

2. A rabbit wants to jump from one side of the road to the other. The rabbit starts at position A and wants to get to B. For simplicity the rabbit can only jump a fixed distance D.
Write an efficient function `int jumps(int A, int B, int D)` that returns the minimal number of jumps required to get from A to B (or further in case the last jump makes the rabbit jump over B).
For example: given A=10, B=85, D=30, the function should return 3. The first jump takes the rabbit to 10+30=40. The second jump to 40+30=70. The third jump to 70+30=100.

3. Integers in their binary representation contain 1s and 0s. For example 1001 in binary is 9 in decimal. Write a function `int zeroes(int i)` that returns the largest number of consecutive 0s surrounded by 1s.
For example given i = 168000 (which is binary 101001000001000000) return 5. The last 6 zeroes are not surrounded by 1s, since there is only a 1 to the left.

4. Our ETL-pipeline handles massive amounts of log files from our Adservers every day. Some years ago we changed our system from a MySQL based one to a Hadoop based one. Since then we write our logs to HDFS.
During a transition phase we had both systems running and therefore produced two log files: one written to MySQL and one to HDFS. Even though the results were supposed to be identical and were describing the exact same events, we sometimes had differences between the two logs.

Travel back in time with us and investigate one of those incidents where our logs differed:
Given are two different log files. One log is a dump out of HDFS (*hadoop.csv*) and the other one of MySQL (*mysql.csv*). Although the logs represent the same events, the number of entries in the logs differ. We suspect that the dump out of HDFS is missing some entries (that are in the dump out of MySQL). The logs differ in format so they cannot be compared directly.
- Find the missing entries and explain us how you did so.
- Find a pattern that will make it easier for us to identify the source of the problem.

5. Set up a single node Kafka cluster and feed it with the MySQL data from task 4. Each line should be a single message. Write a consumer which prints the messages you have just fed into Kafka to stdout.

6. In the BI department we offer data and specifically reports to our users. The users might be other departments or the customer itself that wants to see the amount of advertisements our servers delivered.
Providing data via an API is therefore part of our daily routine. In this task we want you to write a small service that reads out the data from task 4 and exposes it via an API.

- Set up a SQL Server (MySQL or SQLite could be a good choice) with a table that holds the data of the file *mysql.csv*.
- Write a service that accesses the data in your database and returns the following information: Given an userId, the API returns the amount of advertisements the user has seen. One entry in the *mysql.csv* file represents an advertisement delivery.

  For example for the userId 6096033656594890947 the API should return 38.

  HINT: If you know Docker it is probably not a bad idea to use it for this task.