Exercise 1: Hello world

In this exercise you will write your first Reo program that outputs "Hello, world!" on the screen. Assume that your are given 8 different components: a producer component for every character/space in "H", "e", "l", "o", " ", "w", "r", "d", "!".

1. Using pen and paper, design a Reo connector with 9 input ports (one for each character) and a single output port such that every get operation on the output port returns the next character from the string Hello, world!.

2. Implement your designed connector in the Reo language by implementing the following component:

```
connector(a[1..9], b) {
    ...
}
```

3. Create a main component that links the connector with Java components:

```
main() {
   cons(x) prod_H(a) ...
   connector(a, b, c, d, e, f, g, h, i, x)
}

cons(a?String) { #JAVA "Components.cons" }
prod_H(a!String) { #JAVA "Components.prod_H" }
...
```

   where the Java components are defined in Component.java:

```
import nl.cwi.reo.runtime.InputPort;
import nl.cwi.reo.runtime.OutputPort;
public class Components {
   public static void cons(InputPort p) {
      for (int i = 0; i < 10; i++)
         System.out.print(p.getUninterruptibly());
   }

   public static void prod_H(OutputPort p) {
      while(true)
         p.put("H");
   }
   ...
   }
```

Exercise 2: Chess

Design and implement a Reo protocol that coordinates the play between two chess programs and a chess board that displays each move. You can use the files that you get with this exercise.