Exercise 1: Friendship
Consider the following classes.

```java
class LogManager {
    Vector<String> buf;
    public LogManager() {
        buf = new Vector<String>();
    }
    public void add(String msg) {
        buf.add(msg);
    }
}

class LogWriter extends Thread {
    Vector<String> buf;
    int pos;
    public LogWriter(LogManager m) {
        buf = m.buf;
        pos = 0;
        setDaemon(true);
        start();
    }
    public void run() {
        try {
            while (true) {
                Thread.sleep(100);
                if (pos != buf.size()) {
                    String msg = buf.get(pos++);
                    System.err.println(msg);
                }
            }
        }
    }
}
```
catch (InterruptedException ie) {} } } 

These classes are an abstraction of a logging system. The system comprises of a LogManager that can serve multiple LogWriter instances. The desired invariant for the LogWriter class is $\text{pos} \leq \text{buf.size()}$.

(a) Explain why this invariant cannot be established with the ownership model.

(b) Explain how these problems are solved by the friendship model

**Exercise 2: Logical operators**

From the logical operators $\text{false}$, $\rightarrow$ and $\forall$, all other logical operators are definable. For example $\neg F$ can be defined as $\neg F := F \rightarrow \text{false}$. Find formulas defining

(a) $\neg F$

(b) $\text{true}$,

(c) $F \lor G$

(d) $F \land G$

(e) $\exists x F$

in terms of $\text{false}$, $\rightarrow$ and $\forall$. Prove the validity of these definitions in sequent calculus, e.g. $\neg F \rightarrow F \rightarrow \text{false}$ and $F \rightarrow \text{false} \rightarrow \neg F$. 