POPL Assignment 4 (Small)

1. Write a Haskell function **unify** to unify two terms according to the following specifications:

```
unify :: (Term, Term) -> [(Var, Term)]

data Var = U | V | W | X | Y | Z
data Fun = A | B | C | D | E | F | G | H | I | J
data Gtree = Gnode Fun [Gtree] | Leafv Var
type Term  = Gtree
```

The function `unify` should either return the MGU or abort with an appropriate error message (Use `error ‘‘MGU not possible’’` OR `error ‘‘Infinite MGU’’` to report error).

As you can guess, Gtree represents terms, where Gnode represents nodes having k-ary (k ≥ 0) functions and Leafv represents leaf nodes having Variables.

Sample IO (The order of pairs in the output may differ):

#	**INPUT**	**OUTPUT**
1	unify(Gnode G [Gnode A [ ], Gnode H [Leafv Y], Leafv Z], Gnode H [Gnode A [ ], Gnode F [Gnode F [Leafv V]], Gnode B [ ]])	*** Exception: MGU not possible
2	unify(Gnode G [Gnode A [ ], Gnode F [Leafv Y], Leafv Z], Gnode G [Gnode A [ ], Gnode F [Gnode F [Leafv V]], Gnode B [ ]])	[(Z,Gnode B [ ]),(Y,Gnode F [Leafv V])]
3	unify(Gnode G [Leafv X, Gnode H [Gnode A [ ], Leafv Z], Gnode F [Leafv X]], Gnode G [Gnode J [Gnode J [Leafv V]], Leafv Y, Gnode F [Leafv W]])	[(W,Gnode J [Gnode J [Leafv V]]),(Y,Gnode H [Gnode A [ ],Leafv Z]),(X,Gnode J [Gnode J [Leafv V]])]
4	unify (Gnode I [Gnode F [Leafv X], Gnode G[Leafv Y]], Gnode I [Leafv Z, Gnode G [Leafv V]])	[(Y,Leafv V),(Z,Gnode F [Leafv X])]
5	unify (Gnode I [Gnode F [Leafv Z], Gnode G[Leafv Y]], Gnode I [Leafv Z, Gnode G [Leafv V]])	*** Exception: Infinite MGU

(15 Marks)

2. Programmers at the Flaky Computer Corporation designed the protocol shown below to achieve n-thread mutual exclusion.

```
class Flaky implements Lock {
    private int turn;
    private boolean busy = false;
    public void lock() {
        int me = ThreadID.get();
        do {
            do {
                turn = me;
            } while (busy);
            busy = true;
        } while (turn != me);
    }
    public void unlock() {
        busy = false;
    }
}
```

(a) Does this protocol satisfy mutual exclusion? (5 Marks)

(b) Is this protocol starvation-free? (5 Marks)

(c) Is this protocol deadlock-free? (5 Marks)

Give justification for your answers (a proof-sketch, or a counter-example).

The End.