Programming Assignment 2

Date of submission: The 15th-16th November weekend

Write an interpreter for a language, represented by the Haskell data type `Pgm` shown below:

```
type Pgm = (Eqns, Exp)
type Eqns = [Eqn]
type Eqn = (Var, Exp)
data Exp = App Exp Exp | Lam Var Exp | Var Var | Con Con | I Integer
data Var = A | B | C | X | Y | Z | F | G  | H
data Con = TT | FF | Nil | Hd | Tl | Cons | Plus | Minus | IF | EQ
```

1. The input to the interpreter is a syntactically correct program. Any type errors should be trapped at runtime and an appropriate error message generated.
2. The evaluation strategy should be *lazy*. You may want to first implement normal-order evaluation strategy and then, to get full credit, extend it to lazy evaluation.