I'm going to learn some agda!

```
data Greeting : Set where
    hello : Greeting

greet : Greeting
greet = hello
```

Defining the natural numbers:

```
data Nat : Set where
    zero : Nat
    suc : Nat → Nat

{-# BUILTIN NATURAL Nat #-}

_+_ : Nat → Nat → Nat
zero + y = y
suc x + y = suc (x + y)
```

Exercise 1.1 Define the function `halve : Nat → Nat` that computes the result of dividing the given number by 2 (rounded down). Test your definition by evaluating it for several concrete inputs.

```
halve : Nat → Nat
halve 0 = 0
halve 1 = 0
halve (suc (suc n)) = halve n + 1
```

Exercise 1.2 Define the function `_*_: Nat → Nat → Nat` for multiplication of two natural numbers.

```
_*_ : Nat → Nat → Nat
0 * y = 0
suc x * y = y + (x * y)
```

Exercise 1.3 Define the type Bool with constructors true and false, and define the functions for negation `not : Bool → Bool`, conjunction `_&&_ : Bool → Bool → Bool`, and disjunction `_||_ : Bool → Bool → Bool` by pattern matching.

```
data Bool : Set where
    true : Bool
    false : Bool

not : Bool → Bool
not true = false
not false = true
```