Q1. Write one Prolog program that solves the following cryptarithmetic puzzle:

TOO + TOO + TOO = MUCH

In it, each letter represents a different digit from 0 to 9.

Q2. Write a Haskell program using the higher order functions foldl or foldr and the list constructor (:) that emulates the list concatenation function append defined as follows:

```haskell
append [] ys = ys
append (x:xs) ys = x : (append xs ys)
```

Q3. Write a Scala program (possibly using case classes and case objects) equivalent to the following Haskell program:

```scala
data Nat  = Zero | Odd Nat | Even Nat deriving (Eq,Read,Show)

after Zero = Odd Zero
after (Odd n) = Even n
after (Even n) = Odd (after n)

before (Odd Zero) = Zero
before (Odd n) = Even (before n)
before (Even n) = Odd n
```

Test the correctness of your program by checking that, for 3 different values n:

```scala
before (after n) == n
after (before n) == n, for n <> Zero
```

Q4. Develop each of the following 2 topics on not more than 1/2 page each.

- **A:** What are the key differences between Haskell’s type classes, Scala’s abstract classes and traits?
- **B:** Explain the similarities and differences between mark-and-sweep and mark-and-copy garbage collectors. Explain how the marking phase share by both works.