FUNCTIONAL PROGRAMMING 2018

MOISE GABRIEL

QUESTION 2

(a)

```
> zip :: [a] -> [b] -> [(a,b)]
> zip (x:xs) (y:ys) = (x,y) : zip xs ys
> zip   _      _    = []
```

When the function is applied to lists of different lengths zip only creates the first min pairs with the first min elements from list1 and the first min elements from list2, where min = min (length list1) (length list2).

The function is strict in both of its arguments, due to pattern-matching.

(b)

```
> zipWith :: (a -> b -> c) -> [a] -> [b] -> [c]
> zipWith f (x:xs) (y:ys) = f x y : zipWith f xs ys
> zipWith _   _      _    = []
```

(c)

```
> zip' :: [a] -> [b] -> [(a,b)]
> zip' = zipWith (\ x y -> (x,y))

> zipWith' :: (a -> b -> c) -> [a] -> [b] -> [c]
> zipWith' f xs = map (uncurry f) . zip xs
```

These two equation cannot be used together because in order to apply them, we would be stuck in an infinite loop.

zip' would need zipWith' which would mean to apply (uncurry f) to all elements of the list formed by zip' and so on..

"power series" $a = \sum_{i=0}^{\infty} a_i x^i$, represented by $[a_0, a_1, a_2, \dots]$

(d) The power series 0 is [0,0..], an infinite list of zeroes, whereas the power series 1 is [1,0,0..] 1 followed by an infinite number of zeroes.

```
> zero :: [Double]
> zero = 0 : zero

> one :: [Double]
> one = 1 : zero
```

The value of `zero == one` is clearly False, since `(x:xs) == (y:ys) = (x == y) && (xs == ys)` and from the start `0 == 1` is False, whereas `zero == zero` is $\bot$, since we enter an infinite loop.

(e) $$\sum_{i=0}^{\infty} a_i x^i + \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} (a_i + b_i) x^i$$

```
> plus :: [Double] -> [Double] -> [Double]
> plus = zipWith (+)
```

$$\sum_{i=0}^{\infty} a_i x^i - \sum_{i=0}^{\infty} b_i x^i = \sum_{i=0}^{\infty} (a_i - b_i) x^i$$

```
> minus :: [Double] -> [Double] -> [Double]
> minus = zipWith (-)
```

(f) $$\frac{d}{dx} \sum_{i=0}^{\infty} a_i x^i = \sum_{i=1}^{\infty} i a_i x^{i-1}$$

```
> deriv :: [Double] -> [Double]
> deriv = zipWith (*) [0..]
```

$$\int_0^x \sum_{i=0}^{\infty} a_i x^i = \sum_{i=0}^{\infty} \frac{a_i}{i+1} x^{i+1}$$

```
> integral :: [Double] -> [Double]
> integral xs = 0 : zipWith (/) xs [1..]
```

(g) The equation `expx = deriv expx` will define bottom, since `deriv expx = zipWith (*) [0..] expx`, which will require the first element of `expx`, so we'll enter an infinite loop.

The equation ``expx = one `plus` integral expx`` is equivalent to

``[1,0,0..] `plus` (0 : zipWith (/) expx [1..])``

and because of "lazy" evaluation, we first obtain 1.0, then 1.0, then $0.5 = \frac{1}{2!}$, then $0.16 = \frac{1}{3!}$, and so on, as $e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$

(h) Since we have

$$\sin x = \int_0^x \cos x \, dx - 1$$

$$\cos x = -\int_0^x \sin x \, dx$$

Then

```
> sinx :: [Double]
> sinx = integral cosx `minus` one
> cosx :: [Double]
> cosx = zero `minus` integral sinx
```