

Taller de Álgebra 1

Ejercicios

April 8, 2021

1. Escriba una función `dígito :: Int -> Integer -> Integer` tal que la expresión `dígito i n` calcule el *i*-ésimo dígito decimal del entero *n*, asumiendo que *n* es un entero no negativo y que *i* es un entero positivo.

```
*Main> dígito 1 12345
5
*Main> dígito 3 12345
3
*Main> dígito 5 12345
1
*Main> dígito 10 12345
0
```

2. Escriba una función `sumaDeDígitos :: Integer -> Integer` tal que la expresión `sumaDeDígitos n` sea la suma de los dígitos del entero *n*, que asumimos es no negativo.

```
*Main> sumaDeDígitos 12345
15
*Main> sumaDeDígitos 8
8
*Main> sumaDeDígitos 0
0
```

3. Escriba una función `dígitoMáximo :: Integer -> Integer` tal que la expresión `dígitoMáximo n` sea dígito más grande de *n*, que asumimos es no negativo.

```
*Main> dígitoMáximo 123454321
5
*Main> dígitoMáximo 10001
1
*Main> dígitoMáximo 8
8
*Main> dígitoMáximo 0
0
```

4. Escriba una función `factorial :: Integer -> Integer` tal que `factorial n` sea el factorial del entero no negativo *n*

```

*Main> factorial 10
3628800
*Main> factorial 1
1
*Main> factorial 0
1

```

5. Escriba una función `e_approx :: Integer -> Double` tal que para cada entero no negativo `n` el valor de `e_approx n` sea

$$\sum_{k=0}^n \frac{1}{k!}.$$

```

*Main> e_approx 10
2.7182818011463845
*Main> e_approx 2
2.5
*Main> e_approx 1
2.0
*Main> e_approx 0
1.0

```

6. Escriba una función `unos :: Integer -> Integer` tal que para cada entero no negativo `n` el valor de `unos n` sea la cantidad de dígitos 1 en la escritura binaria de `n`.

```

*Main> unos 12345
6
*Main> unos (2^6 - 1)
6
*Main> unos 2^6
1
*Main> unos 0
0

```

7. Escriba una función `pi_approx :: Integer -> Double` tal que para cada entero positivo `n` el valor de `pi_approx` sea

$$3 + \underbrace{\frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \cdots \pm \frac{4}{2n \times (2n+1) \times (2n+2)}}_{n \text{ sumandos}}$$

Observe que los sumandos marcados van alternando en signo — en particular, el signo del último sumando depende de la paridad de `n`.

```

*Main> pi_approx 1000
3.1415926533405423
*Main> pi_approx 10
3.1414067184965018
*Main> pi_approx 1
3.1666666666666665
*Main> pi_approx 0
3.0

```