

Taller de programación en Haskell - Recurrencia

Resolver cada uno de los siguientes ejercicios mediante un programa en Haskell usando recurrencia. Tener en cuenta identificar claramente los casos base y dar la definición recurrente. Investigar sobre los conceptos o definiciones según se considere necesario.

Ejercicio 1. Factorial

Dado: Un entero positivo n .

Calcular: El factorial de n . (El factorial de n se denota por $n!$ y es igual al producto de todos los enteros positivos desde 1 hasta n).

Ejemplos:

```
factorial(1) = 1
factorial(4) = 24
factorial(10) = 3628800
```

Ejercicio 2. Números de Catalan

Dado: Un entero no negativo n .

Calcular: El número de Catalan C_n . ($C_0 = 1$ y $C_n = [2(2n-1)/(n+1)] C_{n-1}$).

Ejemplos:

```
catalan(1) = 1
catalan(5) = 42
catalan(8) = 1430
```

Ejercicio 3. Número inicial

Dado: Un número entero no negativo n .

Calcular: El dígito inicial de n .

Ejemplos:

```
digitoInicial(7) = 7
digitoInicial(67945) = 6
```

- - Bono: Calcular los 3 dígitos iniciales de un número n .

Ejercicio 4. Potencias de 2

Dado: Entero no negativo n (exponente).

Calcular: La potencia 2^n .

Ejemplos:

```
potenciaBase2(0) = 1
potenciaBase2(5) = 32
potenciaBase2(10) = 1024
```

Ejercicio 5. Raíz digital

Dado: Un número entero n , no negativo.

Calcular: La suma de todos los dígitos que componen el número n .

Ejemplos:

```
raizDigital(8921) = 20
raizDigital(340) = 7
```