

Nombre:.....

1. Determinar el tipo de las siguientes funciones

`subst f g x = f x (g x)`

`cross (f,g) (x,y) = (f x, g y)`

2. Da un habitante para el tipo

`(a -> b -> c) -> b -> a -> c`

3. Escribe el tipo de la función

```
f x y z t
| z < t    = x == y
| otherwise = False
```

4. Escribe en haskell la función de Ackermann definida por:

$$A(m, n) = \begin{cases} n + 1 & m == 0 \\ A(m - 1, 1) & m > 0 \& n == 0 \\ A(m - 1, A(m, n - 1)) & m > 0 \& n > 0 \end{cases}$$

5. Escribe la función

`cerosUnos :: Integer -> (Integer, Integer)`

que toma como argumento un número binario y devuelve un par donde la primera componente cuenta el número de ceros que hay en el argumento y la segunda componente cuenta el número de unos.

```
Main> cerosUnos 1010101010000011101
(10,9)
```