



Nombre y Apellido:

Legajo:

Examen

1. Dar almenos dos ejemplos de funciones que tengan los siguientes tipos:

a) `(Bool -> Bool) -> (Int -> Int)`

b) `Bool -> (Int -> Bool)`

2. Dar el tipo de las siguientes funciones:

a) `clima1 x y z = if z y then y +x else x`

b) `clima2 x y f = if y x then head x else tail f`

3. La función `splitCond` toma un predicado (condición) y una lista, retorna una tupla de listas donde la primera componente tiene los elementos que cumplen con el predicado y en la segunda los que no. **Ejemplo:** `splitCond even [1..10] = ([2,4,6,8,10],[1,3,5,7,9])`

Definir la función `splitCond`:

a) recursivamente sin usar funciones predefinidas y sin usar `foldr`

b) usando `foldr` dando la función asociada al `foldr` como una función anónima.

c) usando listas por comprensión sin usar funciones predefinidas, salvo operadores lógicos.

4. Un gestor de proyecto asigna a cada tarea un identificador y un peso. Mientras mayor peso más importante la tarea y se debe realizar primero. Para implementar el gestor se te pide que:

a) Completa la siguiente definición de tipo:

```
< completar > < completar > = Int
< completar > ID = String
< compleatr > Tarea = (ID,Peso)
data Gestor = Vacio | GS < completar > Tarea < completar >
```

b) Si el `Gestor` se implementa como los invariantes de un BST:

- i) Escribir la siguiente lista de tareas `[("Do",5),("Di",2),("Da",3),("De",8),("Du",8)]` con la representación del tipo `Gestor`, suponiendo que las tareas fueron llegando (insertando) de izquierda a derecha en el `Gestor`.
- ii) Definir la función `agenda` que dado un `Gestor` retorna una lista de IDs ordenados del más al menos urgente, si hay empate se hace el que llegó primero.

5. Se quiere que respondas para el tipo `LeftistHeaps`:

a) ¿Qué es una espina en un árbol? ¿Qué es el rango de un `Leftist Heaps`?

b) ¿Cuál es el invariante de Heaps? ¿Cuál es el invariante de `Leftist Heaps`?

c) Nombra las consecuencias que tienen el/los invariantes de `Leftist Heaps`