

Lista de ejercicios preparatorios para el Primer
Examen Parcial de la Materia de Programación
Funcional. 7o. semestre de la Carrera de
ingeniería en computación.

Universidad Tecnológica de la Mixteca.

25 de octubre de 2023

1. Escribe una función que, dado un valor numérico x , obtenga como resultado la tripleta $(x, 4 * x, x * x)$.
2. Escribe una función anónima que junto con la función `map` obtenga los valores al cubo de una lista de números.
3. Escribe una función que devuelva todos los pares ordenados (a, b) tales que a puede tomar un valor entero entre $a1$ y $b1$, b un valor entero entre $a2$ y $b2$, y $a + b == 17$, con $a1$, $b1$, $a2$ y $b2$ enteros positivos.
4. Describa la lista infinita de números pares positivos como una función escrita en Haskell.
5. Traduzca la siguiente expresión en Haskell a una que utilice `where`:

```
Prelude> let ts n=(2*n):(ts (n+1)) in take 10 (ts 30)
[60,62,64,66,68,70,72,74,76,78]
```

6. Compruebe las siguientes propiedades de funciones de alto orden en Haskell mediante diversos ejemplos (aquí el símbolo de igual funciona como la igualdad en matemáticas):

- a) $(f \cdot g) \cdot h = f \cdot (g \cdot h)$
- b) `map (f.g) ls = map f (map g ls)`
- c) `map f (ls ++ ms) = (map f ls) ++ (map f ms)`
- d) $(f \ \$ \ \text{if expression then } m \ \text{else } n) =$
 $\text{if expression then } f \ m \ \text{else } f \ n$
- e) `[f a | a<- ls] = map f ls`
- f) `filter q ls = [a | a<-ls, q a]`

g) Compruebe que esto no es verdad en general:
`concat ms = reverse (concat (reverse ms))`
 pero esto sí:
`concat ms = concat (reverse (reverse ms))`

7. Halle elementos enteros de un árbol binario (con nodos internos y hojas conteniendo enteros) los cuales son pares.
8. De el dato “días de la semana” y defina la función siguiente día (note que ésta función es completamente definida). [Una función se llama completamente definida sobre un dominio si no se indefine en alguno de los elementos del dominio.]
9. Sea la función triplemente recursiva tipo Fibonacci:

```
f 0 = 1

f 1 = 1

f 2 = 1

f n | n > 2 = f(n-3) + f(n-2)+f (n-1)
```

El cálculo de `f 100` es prácticamente imposible con esta versión de esta función. Utilizando listas infinitas, inspirados con el ejemplo de Fibonacci visto en clase, calcule el valor de `f 100`. El resultado tiene que ser:

```
*Main> ftris !! 100

127071617887002752149434981
```

10. Compruebe que:

```
sqr . even = even . sqr
where sqr x = x*x
```

transgrede la noción de tipo (los tipos no casarían).