

**Examen Parcial Introducción a los Algoritmos - 17 de Marzo de 2013**  
**Comisiones Tarde**

Apellido y Nombre:

nota
------

Cantidad de hojas entregadas:

1	2	3	4	5
---	---	---	---	---

1. [15 pto(s)] Decidir si las siguientes expresiones son satisfacibles o insatisfacibles. Si son satisfacibles decidir si son válidas o no. En el caso de que la expresión sea satisfacible, dar valores para las variables que hacen que la expresión sea verdadera. En todos los casos usar los significados usuales de las operaciones involucradas.
  - a)  $x^2 + 1 = 0$
  - b)  $(length\ xs) > 2 * n$  (o  $(\#.xs) > 2 * n$  en notación del libro)
  - c)  $[] ++ xs = xs ++ []$
  
2. [10 pto(s)] Definir la siguiente función.  $multiplo : Int \rightarrow Int \rightarrow Bool$ , que dados dos enteros devuelve *True* si el primero es múltiplo del segundo. Ejemplos:  $multiplo\ 4\ 2 = True$ ,  $multiplo\ 2\ 4 = False$
  
3. (a) [15 pto(s)] Definir la siguiente función recursiva:  $sacaEspacio :: [Char] \rightarrow [Char]$ , que dada una lista de Char devuelve la lista sin espacios.  
 (b) [5 pto(s)] Luego, evaluar manualmente la función para el ejemplo, justificando cada paso. Ejemplo:  $sacaEspacio\ ['s', ' ', 'i'] = ['s', 'i']$ .
  
4. (a) [15 pto(s)] Definir la siguiente función recursiva:  $clasifica :: [Int] \rightarrow [(Int, Char)]$ , que dada una lista de enteros (que representan temperaturas) devuelve la lista de pares que clasifica cada número en "Frio" (usando 'F') o "Caliente" (usando 'C') dependiendo si son menores estrictos que 0 o no.  
 (b) [5 pto(s)] Luego, evaluar manualmente la función el ejemplo dado, justificando cada paso. Ejemplo:  $clasifica\ [1, 0, -5] = [(1, 'C'), (0, 'C'), (-5, 'F')]$ .
  
5. [35 pto(s)] Usando las siguientes definiciones

Haskell	Formalismo Básico
1) $sumal\ [] = []$	$sumal.[\ ] \doteq [\ ]$
2) $sumal\ (x:xs) = (x + 1):sumal\ xs$	$sumal.(x \triangleright xs) \doteq (x + 1) \triangleright sumal.xs$
3) $length\ [] = 0$	$\#.[\ ] \doteq 0$
4) $length\ (x:xs) = 1 + length\ xs$	$\#.(x \triangleright xs) \doteq 1 + \#.xs$
5) $sum\ [] = 0$	$sum.[\ ] \doteq 0$
6) $sum\ (x:xs) = x + sum\ xs$	$sum.(x \triangleright xs) \doteq x + sum.xs$

Demuestre por inducción la propiedad:

$$sum\ (sumal\ xs) = (sum\ xs) + (length\ xs) \mid sum.(sumal.xs) = (sum.xs) + (\#.xs)$$