**TC2006 – Lenguajes de Programación***Evaluación Perezosa y Otras Facilidades en Haskell*

Matrícula1\_\_\_\_\_\_\_\_\_\_\_ Matrícula2 \_\_\_\_\_\_\_\_\_\_\_\_

1. Programar la función **cuenta\_signo** en Haskell usando **guardias** que a partir de una lista de enteros que recibe como argumento, cuente el total de negativos y positivos y los regrese en el par (#negativos, #positivos). El cero no se cuenta.

Probar con:

> cuenta\_signo [-3,2,0,1,-1] => (2,2)

> cuenta\_signo [0,1,2,3] => (0,3)

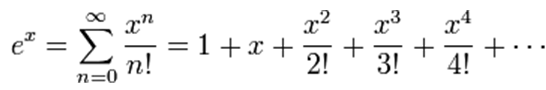
1. Programar la función **separa** en Haskell usando **comprensión de listas** que separe una lista de enteros en dos listas que contengan los valores menores que N en una y los mayores e iguales a N en la otra, dentro de un par.

Probar con:

> separa 0 [-3,2,0,1,-1] => ([-3,-1],[2,0,1])

> separa 2 [0,1,2,3] => ([0,1],[2,3])

1. Programar la función **fexp** en Haskell usando listas infinitas que calcule la función exponencial a partir de N términos de la siguiente serie infinita:



Probar con:

> fexp 4.0 1 => 1  
> fexp 4.0 2 => 5.0   
> fexp 4.0 7 => 48.5555555555556  
> fexp 4.0 28 => 54.598150033144