

INFORMATICA.

EXAMEN FINAL Septiembre 2004.

Haskell

Apellidos, Nombre

Grupo

NOTA: Indica también el tipo de todas las funciones que defines en los distintos ejercicios.

1) (3 puntos) Sea el siguiente tipo para representar números racionales

```
infix 9 :/  
data Racional = Integer :/ Integer deriving Show
```

donde el primer entero es el numerador y el segundo es el denominador. Haz este tipo instancia de las clases *Eq*, *Ord*, *Num* y *Fractional*.

2) (2 puntos) Usando *foldr*, define una función que devuelva las partes (todas las sublistas) de una lista

```
partes [1, 2, 3] => [ [], [1], [2], [3], [1, 2], [2, 3], [1, 3], [1, 2, 3] ]
```

Nota: no es necesario que los resultados aparezcan en el mismo orden que en el ejemplo.

3) Sean los siguientes tipos para representar matrices como listas de vectores (uno por cada fila de la matriz):

```
type Vector = [ Float ]  
type Matriz = [ Vector ]
```

```
m :: Matriz  
m = [ [1,2,3]  
      , [4,5,6]  
      , [7,8,9] ]
```

- a) (2 puntos) Define un operador $\langle * \rangle$ que calcule el producto de dos matrices.
- b) (3 puntos) Define una función `desplaza`, que tome una matriz y devuelva la matriz que se obtiene al desplazar cada elemento una posición adelante. El último elemento de cada fila debe pasar a la primera posición de la fila siguiente. El último elemento de la última fila debe pasar a la primera posición de la primera fila:

```
desplaza [ [1,2,3]  
           , [4,5,6]  
           , [7,8,9] ] => [ [9,1,2]  
                          , [3,4,5]  
                          , [6,7,8] ]
```

INFORMATICA.

EXAMEN FINAL Septiembre 2004.

Pascal

Apellidos, Nombre

Grupo

1) (5 puntos) Consideremos el siguiente tipo para representar una matriz de tamaño 100x100 como máximo:

```
CONST
  MAXIMO = 100;

TYPE
  RANGO = 1 .. MAXIMO;
  MATRIZ = RECORD
    Filas, Cols : RANGO;
    M : ARRAY [RANGO, RANGO] OF REAL
  END;
```

donde las componentes `Filas` y `Cols` almacenan el número de filas y columnas de la matriz y la componente `M` sus elementos.

Define una función `desplaza`, que tome una matriz y devuelva la matriz que se obtiene al desplazar cada elemento una posición adelante. El último elemento de cada fila debe pasar a la primera posición de la fila siguiente. El último elemento de la última fila debe pasar a la primera posición de la primera fila:

Por ejemplo, para la siguiente matriz 3 x 3:

```
1 2 3
4 5 6
7 8 9
```

la función `desplaza` debe devolver ésta otra:

```
9 1 2
3 4 5
6 7 8
```

2) (5 puntos) El método de *Newton-Raphson* para encontrar un cero de una función f de reales en reales, que supondremos derivable, es el siguiente:

- Escoger una aproximación inicial a la raíz, x_0
- Calcular la siguiente aproximación utilizando la fórmula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- Si $|x_n - x_{n+1}| < \epsilon$ entonces x_{n+1} es una raíz. En otro caso volver al punto b).

Escribe una función en Pascal que tome como parámetros una función f , una aproximación inicial x_0 y un valor para ϵ y devuelva un cero de la función usando el método de *Newton-Raphson*. Define también los tipos y subprogramas adicionales que sean necesarios.