

INFORMATICA.

EXAMEN FINAL Junio 2003.

Pascal

Apellidos, Nombre

Grupo

1) Escribe un programa que realice las siguientes tareas:

- Lea los datos de un fichero binario que contiene un máximo de 100 n° reales y los guarde en un vector (0.75 puntos)
- Ordene el vector utilizando el método de inserción (2.5 puntos)
- Escriba un fichero binario con los datos ordenados (0.75 puntos)

Los nombres de los ficheros de entrada y salida se leerán del teclado.

Notas:

- Escribe subprogramas para cada una de estas tareas
- Observa que el número de elementos en el fichero de entrada es variable, aunque puedes estar seguro de que no hay más de 100.

2) (4.5 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.

3) (1.5 puntos) Define los tipos en Pascal necesarios para representar con un registro variante la información referente a un empleado, teniendo en cuenta que para cada empleado se desea almacenar su nombre, sus dos apellidos, su edad y su estado civil (soltero, casado). Si el empleado está soltero se desea almacenar también su dirección, mientras que si está casado se desea almacenar el nombre de la esposa y el número de hijos.

INFORMATICA.

EXAMEN FINAL Junio 2003.

Haskell

Apellidos, Nombre

Grupo

1) (3.5 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:

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

- La primera aproximación x_i tal que $f(x_i) \cong 0$ es el cero buscado.

Escribe una función en Haskell que tome como parámetros una función f , una aproximación inicial x_0 y devuelva un cero de la función usando el método de *Newton-Raphson*. Define también las funciones y operadores adicionales que sean necesarios.

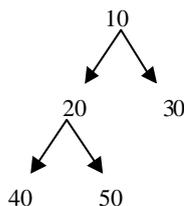
2) (3.5 puntos) Define una función coseno que devuelva una lista infinita con los términos de la siguiente serie:

$$\text{coseno}(x) = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \dots$$

Usa la función anterior y *takeWhile* para calcular una aproximación a *coseno*(0) sumando todos los términos cuyo valor absoluto sea mayor a una diezmilésima.

3) a) (1.5 pts) Una rama de un árbol es una lista con todos los nodos desde la raíz a una hoja del árbol. Define una función *ramasB*, que tome un árbol binario y devuelva una lista con todas sus ramas.

Por ejemplo, para el siguiente árbol



$\text{ramasB } a \implies [[10,20,40], [10,20,50], [10,30]]$

b) (1.5 pts) Define la función *ramas* similar a la anterior pero para árboles generales.