Fundamentos de Programación. Convocatoria de Febrero
Curso 2010/2011
8 de Febrero de 2011

Tiempo: dos horas y media.
IMPORTANTE: Los algoritmos han de ir correctamente explicados.

1. (2.5 puntos) El método de *bisección* nos ofrece un procedimiento para calcular de forma aproximada el punto de corte (la raíz) de una función f con el eje de abscisas.

   Hipótesis de partida: Sea una función real $f(x)$, estrictamente monótona en $[i, d]$, donde $f(i)$ y $f(d)$ tienen distinto signo y sea ϵ una constante real pequeña (del orden de 10^{-4}, por ejemplo).

   Objetivo: Calcular la raíz de f en el intervalo $[i, d]$

   Método de bisección:

   Repetir mientras la diferencia entre i y d sea mayor que ϵ
   Calcular m, el punto medio entre i y d
   Si $f(m)$ es cero, terminar
   Si $f(m)$ tiene igual signo que $f(i)$, cambiar i por m
   Si $f(m)$ tiene igual signo que $f(d)$, cambiar d por m
   El valor aproximado de la raíz de la función f es el valor de m

   Implemente una función **recursiva** que devuelva la raíz de la función f. *Se supone que se dispone de la función f(x) ya implementada en C++.*

2. (2 puntos) Desarrolle un programa para localizar una cadena dentro de otra. El programa leerá dos cadenas desde la entrada estándar, buscará la primera en la segunda, y escribirá en la salida estándar la posición donde se encuentra -en caso de encontrarla- o un mensaje indicando que no se ha localizado. Un ejemplo de ejecución es el siguiente:

   ```
   Introduzca la primera cadena: mundo
   Introduzca la segunda cadena: Hola mumumundo
   "mundo" Se encuentra en la posición 9  de "Hola mumumundo"
   ```

   Tenga en cuenta que no podrá usar ningún método de la clase *string* salvo el acceso a cada uno de los caracteres de las cadenas y al tamaño de las cadenas. En particular, no se pueden utilizar los métodos `find` y `rfind`.

3. Defina la clase `MatrizEnteros` para poder trabajar con una matriz de enteros, de forma que todas las filas tengan el mismo número de columnas.

   a) (0.25 puntos) Definir un método para añadir una fila completa (se añade al final, por lo que la nueva fila pasará a ser la última) y otro método para acceder a una componente i, j de la matriz.

   b) (1.75 puntos) Definir un método para insertar una fila completa en una posición (de fila) determinada. Por ejemplo,

   $$\begin{pmatrix} 3 & 1 & 4 & 5 & 6 \\ 9 & 0 & 4 & 3 & 2 \end{pmatrix} \rightarrow \begin{array}{c}\text{Insertar} \\ (8,5,0,1,2) \\ \text{en la fila 1}\end{array} \rightarrow \begin{pmatrix} 3 & 1 & 4 & 5 & 6 \\ 8 & 5 & 0 & 1 & 2 \\ 9 & 0 & 4 & 3 & 2 \end{pmatrix}$$

   Se quiere que el módulo sea *robusto*, por lo que deben comprobarse las precondiciones pertinentes.

   c) (1.5 puntos) Definir un método que construya y devuelva una matriz con los datos traspuestos de la primera, es decir,

   $$\text{matriz traspuesta}_{ij} = \text{matriz original}_{ji}$$

   d) (2 puntos) Definir un método para sumar aquellas componentes m_{ij} de la matriz m que verifiquen que $i+j$ sea igual a:

   $$\sum_{h=1}^{k} h^2$$

   para algún valor de $k \geq 1$. Por ejemplo, los primeros valores de la serie son $1, 5, 14, \ldots$ Por tanto. habría que sumar las componentes cuyos índices sumen 1, 5, 14, etc, como por ejemplo, $m_{01}, m_{10}, m_{05}, m_{50}, m_{14}, m_{41}, \ldots$