

Tiempo: dos horas y media.

IMPORTANTE: Los algoritmos han de ir correctamente explicados.

1. (2.5 puntos) El método de *bisección* usado para calcular de forma aproximada el punto de corte de una función f con el eje de abscisas (la raíz de la función) se puede enunciar como sigue:

Sea $f(x)$ una función real, 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).

- Calcular m , el punto medio entre i y d ,
- Si $|i - d| < \epsilon$, terminar.
- Si $f(m)$ tiene igual signo que $f(i)$, repetir considerando el intervalo $[m, d]$
- Si $f(m)$ tiene igual signo que $f(d)$, repetir considerando el intervalo $[i, m]$

Implemente una función **recursiva** que reciba como datos de entrada los extremos i, d del intervalo, así como el valor de ϵ y devuelva la raíz de la función. 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 5 de "Hola mundo"
```

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 puede utilizar el método *find*.

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.

- (0.25 puntos) Definir un método para añadir una fila completa y otro método para acceder a una componente i, j de la matriz.
- (1.75 puntos) Definir un método para insertar una fila completa en una posición (de fila) determinada. Por ejemplo,

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

dónde ? representa un valor no utilizado. Se quiere que el módulo sea *robusto*, por lo que deben comprobarse las precondiciones pertinentes.

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

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

- (2 puntos) Definir un método para sumar aquellos valores M_{ij} de la matriz M que verifiquen que $i + j$ sea igual a:

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

para algún valor de $k \geq 1$.