Práctica 5: Introducción a la programación en C++ y al uso de la librería científica gsl

1. Escribir un programa en 'C++' con la misma funcionalidad que el del Ejercicio 11 de la práctica de 'C' pero donde (usando la capacidad de sobrecarga de funciones en 'C++') las funciones homónimas puedan recibir como argumento punteros, además de arreglos de números en doble precisión, a arreglos de enteros y de números con simple precisión (float).

2. Escribir un programa en 'C++' que permita calcular numéricamente

$$\int_1^{10} \ln(x)\, dx$$

   a) Por el método de los trapecios.

   b) Por el método de Simpson.

   En ambos casos, emplear una partición uniforme de N intervalos. Generar gráficos que muestren como varía con N el error respecto del resultado exacto.

3. El archivo 'MuestreoCurvaXY.dat' contiene una tabla de valores (con los valores de x ordenados de manera creciente) correspondiente al muestreo de cierta función $f(x)$, con $a \leq x \leq b$. Escribir un programa en 'C++' que lea ese archivo y, a partir de los datos:

   a) estimar el valor de la longitud del arco de curva.

   b) estimar $\int_a^b f(x)dx$.

4. Hacer un programa en 'C++' que permita aproximar con un error, menor a 10^{-6}, la raíz de $f(x) = e^x - x - 2$ con $x \in [0, 2]$ usando un método de bisección.

5. Escribir un programa en 'C++' en el que se defina una clase llamada 'Elipse' cuyos datos privados son las coordenadas del centro y valores de los semiejes y contenga métodos (funciones miembro públicas) para redefinir las coordenadas del centro y los valores de los semiejes, para imprimir estas cantidades y para imprimir el área encerrada. Realizar un programa que ejemplifique el uso de los objetos de la clase creada.

6. Generar la versión en 'C++' del programa desarrollado en el ejercicio 4 de la práctica de 'C', con la diferencia que la nueva versión genera el archivo de salida e invoca internamente a *xmgrace. Sugerencia: usar la clase 'string' de la biblioteca estándar de C++ para manipular las cadenas de texto.*

7. Mediante los generadores de números aleatorios provistos por la librería gsl, generar un programa en 'C++' que reciba como argumento un número entero N e imprima en pantalla N números aleatorios con una distribuidos

con una distribución que puede ser de dos tipos : normal y lognormal (el tipo de distribución se debe poder elegir estáticamente mediante una directiva del preprocesador). Mediante grace, generar una imagen con un ejemplo de la salida del programa para cada una de las distribuciones en forma de histograma.

8. Mediante las funciones provistas por la librería gsl, generar un programa 'C++' que permita muestrear en N puntos equiespaciados del intervalo $[a, b]$ (los parámetros a, b y N podrán pasarse desde línea de comandos), las siguientes 'funciones especiales' (generar un gráfico ejemplificando cada caso):

   a) La función de Bessel $J_0(x)$ de primera especie de orden 0.

   b) La función error $erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$.

   c) La función de Hermite de orden 4 $\Psi_4(x)$.

9. Mediante funciones provistas provistas por la librería gsl, generar un programa en 'C++' que encuentre una función spline que pase por los puntos $(-4, 10)$, $(-2, -6)$, $(0, 2)$, $(3, -1)$, $(5, 8)$ (generar un gráfico con los puntos y con la curva spline).

10. Resolver analíticamente e implementar tal solución en un programa de 'C++', el sistema lineal $Ax = b$ donde $A \in \mathbb{R}^{N \times N}$ es una matriz no-singular en la que sus únicos elementos no nulos están sobre la diagonal principal y la diagonal inmediata superior, que están dados por $A_{i,i} = D_i, \quad i = 1, 2, \ldots, N$ $(D_i \neq 0, \forall i)$ y $A_{j,j+1} = U_j, \quad j = 1, 2, \ldots, N-1$. Emplear el programa para resolver el sistema lineal

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ -3 \\ 4 \\ 0 \end{pmatrix}$$

    Verificar la solución obtenida empleando *Octave*.

11. Implementar un programa de 'C++' que permita resolver de manera aproximada el sistema lineal del ejercicio 10 mediante el método iterativo de Jacobi. Generar un gráfico que muestre como varía el error (definido como la norma del vector diferencia entre la solución aproximada y la exacta) con el número de iteraciones N.