Ejemplo 3.C

declarar cuadrado como void que con un argumento de entrada que será la variable flotante. →

Abrir "main", declarar las dos variables.

↓

Pedir y escanear la primer variable. Después, nombrar la función dentro de "main".

↓

hacer que "main" no regrese nada (return 0;)

↓

Cerrar "main" y abrir "void cuadrado(float x)" y declarar la variable 2.

↓

en "cuadrado" realiza la operación x2 = x*x, e imprimir el resultado.

Ejemplo 4.C

Antes de main, declarar las dos variables.

↓

Declarar la función "cuadrado" como double sin argumentos de entrada.

↓

abrir "main" e imprimir la segunda variable, la cual tendrá el cuadrado de la primer variable.

↓

Cerrar "main" sin un return.

↓

abrir la función cuadrado, pedir y escanear un número.

↓

asignar a la variable dos el valor del cuadrado de la primera; de la siguiente manera

x2 = x*x

→ Imprimir solo la primera variable

↑

Con return, hacer que devuelva la segunda variable.

Funciones.C

Escribir la librería math y stdio →

el pedazo
Escribir de código (argumento de entrada) de cada función fuera del int →

abrir un int main() y cerrarlo con return 0;

Crear un switch dentro de main para que el usuario decida qué de función quiere usar.

escanear una variable y crear 4 casos para el switch

Dentro de cada caso irán los pedazos de código de cada función, dependiendo de cómo están organizados y si tengan o no argumentos de salida

Para sinh solamente escribir en el caso 1 "sinhy()"

Para cosh escribir que le pregunte al usuario la variable, la escanee y la función coshy(x).

Para cosh, fuera de "main", declarar void coshy(float x), con argumento de entrada, luego declarar la variable que tomará el valor del cosh de la otra variable. Realizar la ecuación para dicha variable, e imprimir ésta.

Fuera de "main", abrir la función sinhy(), declarar las dos variables, pedir y escanear la variable, después calcular el sinh de la variable, y finalmente, imprimir dicho valor, el cual fue asignado en la función sinhy()

Fuera de "main", escribir "void tanhy()". Pedir y escanear un número, luego, con otra variable darle el valor que tomará la primera usando la ec de la tanh, después, hacer que en return, de el valor de la variable que tomó el valor de la primera al ser operada en tanh.

Finalmente, en la función, escribir "float raiz(float x)" y en return, hacer que devuelva el argumento de entrada de la función

Para la tanh, en el caso 3, solamente se imprimirá el resultado por lo que se escribe y se cierra el caso 3.

Para la raíz, pedir al usuario el número, escanearlo, escribir z = raiz(x), para ejecutar la función y darle un valor, y después imprimir el valor del resultado de haber aplicado la función.

Desprom.C

declarar un pointer float, en función de un cierto número de datos. →

Evaluar si dicho pointer es nulo, es decir, si no tiene dirección

Abrir el archivo en el que se encuentran los datos →

Introducir un for para que escanee los datos, y crear una variable que los sume

Fuera del for, dividir la variable que contiene la suma de los datos entre el num de datos

Abrir de nuevo el archivo, escanear los datos (con un for también) y sumarlos en un nuevo sumatorio, pero éste sumará los datos después de elevar al cuadrado la resta del dato con el promedio →

Calcular la raíz del nuevo sumatorio, pero antes dividir a éste entre el número de datos menos 1

Imprimir la media aritmética (sumatorio1 /n) y la desviación estándar (la raíz del sumatorio2 entre (n-1)