

Ejercicios del Tema 6

Funciones

1. Áreas de figuras geométricas

Escribe un programa donde el usuario pueda seleccionar entre calcular el área de un círculo, un triángulo o un cuadrado. El programa empleará una función diferente definida para cada caso. Valgan como ejemplo los siguientes prototipos:

```
float aCirc (float r);  
float aCuad (float l);  
float aTri (float b, float h);
```

2. Conversión de temperaturas

Escribe un programa para realizar la conversión de temperaturas. El programa pide al usuario que introduzca un valor (número real) y una letra. La letra indica la escala en la que se introduce esa temperatura. Si la letra es 'C', la temperatura se convertirá de grados centígrados a Fahrenheit. Si la letra es 'F' la temperatura se convertirá de grados Fahrenheit a grados Centígrados. El programa se repetirá indefinidamente hasta que el usuario introduzca una letra diferente a 'C' o 'F' como escala. Se usarán 2 funciones `cent2fahr` y `fahr2cent` para convertir de una escala a otra. Estas funciones aceptan un parámetro (la temperatura en una escala) y retornan el valor en la otra escala. La relación entre ambas escalas es $T_F = 9/5 * T_C + 32$

3. Tablas de multiplicar

Construye un un programa que muestre por pantalla las tablas de multiplicar del 1 al 10 en base a dos funciones específicas. La primera función debe devolver el producto de dos valores numéricos enteros dados como parámetros. La segunda función debe mostrar por pantalla la tabla de multiplicar de un número dado como parámetro.

4. Distancia entre puntos

Realiza un programa para calcular la distancia entre dos puntos bidimensionales introducidos por el usuario, empleando una función con el siguiente prototipo:

```
float dist (float x1, float y1, float x2, float y2);
```

5. Máximo de 3 números

Escribe un programa que calcule y muestre en pantalla el máximo de tres valores de tipo entero que se introducen por teclado. Este programa usa una función `max3` que admite como parámetros tres números enteros y entrega el valor del mayor. Esta función usará a su vez a otra función `max2` que calcula el máximo entre 2 valores que se pasan como parámetros a la función.

6. Potencia

Realizar un programa que calcule con una función llamada potencia, el valor de a^b , donde a y b son dos números enteros introducidos desde el teclado tal que $a > 0$ y $b \geq 0$ (se supondrá que el usuario introduce dos números que cumplen estas condiciones). Nota: No se pueden utilizar funciones de la librería de math.h.

7. Números combinatorios

Escribe un programa que calcule y muestre en pantalla el número combinatorio a partir de los valores n y k introducidos por teclado.

$$n k = \frac{n!}{(n-k)! \cdot k!}$$

Este programa debe estar construido en base a dos funciones, una para calcular el factorial de un número, y otra para calcular el número combinatorio.

8. Serie

Calcular los n primeros términos impares de la serie:

$$a_i = \frac{i+2}{10}, i = 1, 2, \dots, n$$

a través de una función con el prototipo dado al final del enunciado, siendo n un número entero que se solicita al usuario desde el main() hasta que el usuario teclee un número mayor que cero.

float TerminosN (int n);

9. Serie de Taylor

Escribe un programa que calcule la aproximación de e^{-x} mediante el desarrollo de Taylor:

$$e^{-x} = 1 + \sum_{i=1}^N \frac{(-x)^i}{i!}$$

El programa estará construido en base a tres funciones: factorial calcula el factorial de un número entero n; potencia calcula la potencia n de un número real x; exponencial calcula la aproximación anterior de un número real x usando N términos de la serie de Taylor. Sus prototipos son:

```
float exponencial(float x, int n);  
double potencia(float x, int n);  
long int factorial(int n);
```

El programa solicitará al usuario en main el valor del número real y del número de términos deseados. No se debe utilizar la librería math.h.

10. Fibonacci

Escribe un programa que genere los n primeros términos de la serie de Fibonacci **usando una función recursiva**. El número entero n deberá ser leído por teclado. Este valor debe ser positivo, de forma que si el usuario introduce un valor negativo el programa volverá a pedir que lo introduzca.

En la serie de Fibonacci los dos primeros números son 1, y el resto se obtiene sumando los dos anteriores: 1, 1, 2, 3, 5, 8, 13, 21, ...