

FUNDAMENTOS DE PROGRAMACIÓN

Problemas de los TEMAS 4 y 5: Punteros y Funciones

1. Escribe las siguientes funciones:
 - a) **int intervalo(...)** que a partir de tres números enteros: num, n1 y n2, devuelva al programa de llamada un 1 si num está comprendido entre n1 y n2 y devuelve un 0 en caso contrario (suponemos $n1 < n2$).
 - b) **int fechaCorrecta (...)** que a partir de dos números enteros que representan día y mes de una fecha devuelva 1 si la **fecha es correcta** y 0 si es incorrecta. Se tendrá que utilizar una función que a partir de un mes devuelve el número de días que tiene ese mes, su prototipo es:
`int ultimoDia (int mes);` //esta función hay que utilizarla pero **no hay que codificarla**
 - c) ... **escribeHora(int hh, int mm)** que a partir de dos valores hh y mm que representan horas y minutos, imprima la hora en formato hh : mm
 - d) **int numCifras(...)** que a partir de un número entero positivo o nulo, devuelva la cantidad de cifras del número.
2. Escribe una función que a partir de un número entero positivo mayor que 1, devuelva el **mayor divisor** del número distinto de él mismo.
3. Escribe un programa que lea un número entero positivo y si el número tiene tres cifras escriba su inverso, se deberá utilizar una función que a partir de un **número de tres cifras** retorne su **inverso**. Ejemplo, si el número es 345, la función devolverá el valor 543
4. Realiza los siguientes apartados:
 - a) Escribe una **función** que a partir de un número entero positivo devuelva la suma de sus cifras.
 - b) Escribe un **programa** que calcule e imprima los **10 primeros números** que cumplan que la suma de sus cifras es mayor que 10, utilizando la función del apartado a).
 - c) Escribe un **programa** que calcule e imprima los números **comprendidos entre 200 y 300** que cumplan que la suma de sus cifras sea 9, utilizando la función del apartado a). Modificar el programa para que lea dos valores n1 y n2, e imprima todos los números que cumplan la condición pedida y que estén comprendidos entre n1 y n2
5. Se introducen por teclado una serie de números enteros positivos, el final de la secuencia se marca con un cero. Escribe un **programa** que imprima **si existe** el primer **número perfecto** de la secuencia de entrada. Para resolver el problema se deberán utilizar las siguientes funciones
 - a) **Función Divisores** que calcule y devuelva la suma de los divisores de un número excluido él mismo.
Por ejemplo:
Si introducimos: 30 devolverá 42 ($42 = 1 + 2 + 3 + 5 + 6 + 10 + 15$)
Si introducimos: 6 devolverá 6 ($6 = 1 + 2 + 3$)
 - b) **Función numPerfecto** que indique si un número es perfecto o no lo es. Un número es perfecto, si es igual a la suma de sus divisores (no se incluye él mismo). Se deberá utilizar la función Divisores del apartado anterior.
Por ejemplo: el número 6 es perfecto porque $6 = 1 + 2 + 3$
6. Escribe un **programa** que imprima **todos los números perfectos** que hay en un intervalo. El intervalo está formado por dos números enteros positivos introducidos por el teclado. Se deberá utilizar los **subprogramas** del ejercicio anterior.

7. Disponemos de un fichero de texto **entrada.txt**, formado por palabras separadas entre sí por uno o más caracteres blancos y signos de puntuación.

Escribe un programa C que **crea** un nuevo fichero de texto, **mayusculas.txt**, con el mismo contenido que **entrada.txt** pero con todas las letras en mayúsculas.

Se deberán utilizar dos funciones:

- Función que partiendo de un carácter alfabético minúscula, devuelva la misma letra pero en mayúscula. El prototipo de la función es:

`char pasarAMayusc (char c);`

- Función que partiendo de un carácter, devuelva un 1 si el carácter es una letra minúscula y 0 en caso contrario. El prototipo de la función es:

`int esMinuscula (char c);`

8. Dada una secuencia de números enteros positivos acabada en un entero negativo, escribid un programa que imprima el **primer** número **capicúa** de la secuencia. Para ello realizad una **función** que determine si un número entero positivo es **capicúa**.

9. Se introducen por teclado números enteros positivos, el final de la secuencia es un cero (valor centinela). Escribid un programa que imprima el primer número mayor que 10 cuyas últimas dos cifras sean iguales.

Se deberá codificar y utilizar una función ***void ultimosDigitos (...)***, que a partir de un número entero positivo devuelva la última y la penúltima cifra del número. Ejemplo para num= 2345 devolverá dos valores enteros: 5 y 4

10. Se introducen por teclado una serie de números enteros positivos, el final de la secuencia se marca con un cero. Escribid un programa C que nos indique cuántos números de tres cifras capicúas hay en la secuencia.

Es necesario utilizar **una función (de tipo void)** que partiendo de un número entero positivo, devuelva: la cantidad de cifras, la primera y la última cifra del número.

11. Escribe un programa que presente un menú con tres opciones:

- 1.- dibujar Árbol
- 2.- dibujar Regalo
- 3.- salir del menú

En cada una de las opciones se realizará las siguientes tareas:

Opción 1.:

- leer del teclado un número *num*, entero positivo e impar.
- codificar y llamar a la función ***void arbol(...)*** que visualizará la siguiente figura: (ejemplo para num=7)

```
*
***
*****
*****
```

- volver al menú inicial.

Opción 2.:

- leer del teclado dos números enteros positivos, *base* y *altura*
- codificar y llamar a la función ***void regalo(...)*** que visualizará la siguiente figura: (ejemplo para base=7 y altura=3)

```
*****
*****
*****
```

- volver al menú inicial.

Opción 3.- Salir del menú y finalizar el programa.