
 <p>Universidad Carlos III de Madrid</p>	<p>Departamento de Informática Grado en Ingeniería Informática Sistemas Operativos</p> <p>Clases grupos reducidos 1ª, 2ª y 3ª semana de clase</p>	
-------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

Problema 1:

Escribe el programa “Hola mundo” que tienes en la transparencias de teoría, compila y ejecuta el mismo para familiarizarte con el entorno de trabajo de C.

Problema 2:

Modifica el programa anterior para leer del teclado un nombre y dicho nombre es Juan, tu programa sacará por pantalla “Hola Juan”.

Problema 3:

Realiza un programa C que solicite al usuario dos números, y muestre por pantalla el resultado de sumar dichos números. Se realizarán las siguientes funciones:



- `leerint (int *a);`
Lee un número por teclado y lo devuelve por referencia
- `int sumar (int a, int b);`
Recibe dos números enteros y devuelve su suma
- `void dividir (int a, int b, int *coc, int *resto);`
recibe dos números enteros y devuelve por referencia el cociente y el resto de la división de esos dos números enteros.

El programa principal llamará a estas funciones e imprimirá por pantalla el resultado de realizar las operaciones suma y división.

Problema 4:

Realizar un programa en C que implemente una calculadora de quebrados. Se deben leer dos quebrados por teclado, es necesario definir una estructura para guardar el numerador y el denominador. Hay que imprimir en pantalla un menú que muestre las siguientes opciones: suma, resta, multiplicación y división de quebrados, y por último la opción Salir. La calculadora se ejecutará indefinidamente hasta que se elija la opción Salir. Se deberán realizar las siguientes funciones:

- `int menu(void);`
Muestra por pantalla un menú con las opciones: 1. Suma de quebrados, 2. Resta de quebrados, 3. Multiplicación de quebrados, 4. División de quebrados, 5. Salir. Se solicitará al usuario que elija una de las opciones y la devolverá al programa principal.
- `void Leer_numero(struct quebrado *q);`
Solicitará al usuario los dos enteros que forman parte del quebrado y devolverá dicha lectura por referencia en la variable *q.
- `struct quebrado suma (struct quebrado q, struct quebrado p);`
Recibe como argumento dos quebrados, realiza la suma de los mismos y devuelve el resultado.
- `struct quebrado resta (struct quebrado q, struct quebrado p);`
Recibe como argumento dos quebrados, realiza la resta de los mismos y devuelve el resultado.
- `struct quebrado multiplicacion (struct quebrado q, struct quebrado p);`

 <p>Universidad Carlos III de Madrid</p>	<p>Departamento de Informática Grado en Ingeniería Informática Sistemas Operativos</p> <p>Clases grupos reducidos 1ª, 2ª y 3ª semana de clase</p>	
-------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

Recibe como argumento dos quebrados, realiza la multiplicación de los mismos y devuelve el resultado.

- struct quebrado division (struct quebrado q, struct quebrado p);
Recibe como argumento dos quebrados, realiza la división de los mismos y devuelve el resultado.
- void reducir (struct quebrado q);
Recibe como argumento un quebrado y lo reduce a su fracción canónica (ejem 15/6 pasaría a 5/2.)

Realizar un fichero .h con las cabeceras de las funciones y un fichero .c con el código de las funciones sin incluir el programa principal. El programa principal deberá realizarse en el fichero mainquebrado.c

Problema 5:

Realizar un programa en C que lea por pantalla una cadena de caracteres y cuya ejecución tenga un resultado como el mostrado en el siguiente ejemplo:

Escribe una cadena de caracteres:

El perro de San Roque no tiene rabo porque Ramón Ramírez se lo ha cortado (esta es la frase que introduce el usuario)

Mayúsculas: EL PERRO DE SAN ROQUE NO TIENE RABO PORQUE RAMON RAMIREZ SE LO HA CORTADO

Minúsculas: el perro de san roque no tiene rabo porque ramón ramírez se lo ha cortado

Palíndromo: Odatroc ah ol es zerímaR nónmaR eugrop obar eneit on euqoR naS ed orrep lE



Palabras:

El
perro
de
San
Roque
no
tiene
rabo
porque
Ramón
Ramírez
se
lo
ha
cortado

Número de r: 5

Se leerá en el programa principal la cadena de caracteres y se escribirán las siguientes funciones:

- Una función que pase a mayúsculas todos los caracteres de la cadena

 <p>Universidad Carlos III de Madrid</p>	<p>Departamento de Informática Grado en Ingeniería Informática Sistemas Operativos</p> <p>Clases grupos reducidos 1ª, 2ª y 3ª semana de clase</p>	
-------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

- Una función que pase a minúsculas todos los caracteres de la cadena
- Una función que realice el palíndromo de la cadena de caracteres (escribe todos los caracteres al revés)
- Una función que escriba una palabra por línea.

Problema 6:

Realizar un programa que utilizando las funciones de quebrados del ejercicio anterior permita pedir pares de quebrados por pantalla (hasta que el usuario no desee introducir más) y almacenen en un fichero los pares de quebrados recibidos y la suma de cada par.

Problema 7:

Realizar un programa que lea, como máximo, 10 de los quebrados escritos por el programa anterior en el fichero y los almacene en un array y los muestre ordenados.