1. Determinar el valor de las siguientes expresiones y las conversiones de datos que se hacen de manera automática. Suponer todas las variables con sus valores iniciales para la evaluación de las expresiones.

```
int ia;                         Inicializaciones
int ib;                         ia = 3;
int ic;                         ib = 4;
double da;                      da = 5.2;
double db;                      db = 4;
double dc;
```

```
a) ic = ia /ib;                                          f) ia = ib % 2;
b) da = ia /ib;                          g) (ia == ib) || (da < (db + ib))
c) ic = da /db;                          h) (ia == ib) && (da < (db + ib))
d) dc = da /db;
e) da = ((int) (da / db)) + ib;
```

2. Supongamos que necesitamos dos variables para almacenar la posición del ratón en la coordenada X y en la coordenada Y de la pantalla. ¿Qué identificadores emplearía para cada una de estas variables?

3. ¿Cuáles de los siguientes identificadores (nombre de variables) son incorrectos y por qué?

```
num_camiones
2_registro
_asm
_código
bandera7
num_dia_semana
#codigo
año
_pulsacion
cod soft
```

4. Se han declarado las siguientes variables:

```
int a;  int b;
float c;
```

Completa el especificador de formato de las siguientes instrucciones:

```
printf("El valor de a es %.....", a);
printf("Introduce el valor de b: ");
scanf("%.....", &b);
printf("La suma de a + b es %....", a+b);
printf("Introduce el valor de c:");
scanf("%....", &c);
```

5. Escribir un programa que pida una nota de un alumno, comprobando que está entre 0 y 10, y escriba las calificaciones correspondientes de acuerdo con el siguiente criterio:

```
0 a < 5.0  Suspenso
5 a <7.0   Aprobado
7.0 a <9.0  Notable
9.0 a <=10  Sobresaliente
```

6. ¿Cuál sería el valor de la variable contador después de ejecutarse el siguiente código?

```
int contador = 0;
contador++;
contador = contador + 1;
contador = 1 + contador;
contador--;
contador = 7 - contador;
contador = contador - 4;
```

7. Escribir un programa completo (cabecera, includes, ...) que realice las siguientes operaciones:
   - Declarar dos variables reales x e y.
   - Pedir al usuario el valor de x mediante scanf. Obligar a que x sea positivo
   - Calcular la expresión y = log(3x3)
   - Mostrar los valores finales de x e y

   Nota: para calcular el logaritmo neperiano se usa la función log(), lo cual requiere añadir al principio del programa el include: `#include <math.h>`

8. ¿Cuál sería el resultado de la siguiente expresión?

```
! ( (7<=7) || (33 > 33) )
```

9. Escriba un programa que crea un array de tamaño 100 y lo rellene con los valores 10, 20, 30,... de diez en diez. Finalmente, imprima el array en la pantalla. No hay que usar funciones.

10. Realice un programa que almacene en un array unidimensional de 10 letras escritas por el usuario. A continuación, se debe mostrar un mensaje en pantalla indicando cuántas vocales se escribieron.

11. Escriba un programa que pida una letra al usuario y luego llame a una función que imprima el carácter siguiente en la tabla ASCII.

12. Escriba un programa que pida un número al usuario y luego llame a una función que imprima el cuadrado de dicho número.

13. Los números de Fibonacci forman una interesante secuencia en la que cada número es igual a la suma de los dos números anteriores.

$$F_1 = F_2 = 1$$
$$F_i = F_{i-1} + F_{i.-2}$$

Realizar la función, **`FibonacciRec()`** (recursivo) que recibe como argumento el número de Fibonacci que se quiere calcular y devuelvan dicho valor. Ejemplo:

$F_0 = 0$
$F_1 = 1$
$F_2 = 1$
$F_3 = F_1 + F_2 = 1 + 1 = 2$
$F_4 = F_3 + F_2 = 2 + 1 = 3$

14. Suponiendo que un programa ha leído un valor entero mínimo `min` y un valor entero máximo `max`, escribir un bucle `for` que muestre por pantalla una tabla de valores y valores al cuadrado para todos los números comprendidos entre `min` y `max`.
    Ejemplo: si `min` vale 9 y `max` vale 12 el resultado sería

```
x       x*x
---------------
9       81
10      100
11      121
12      144
```

15. Escribir un programa completo (cabecera, includes, ...) que realice las siguientes operaciones:
    - Declarar una variable `x` (tipo `double`) y otra `n` (tipo `int`).
    - Leer el valor de ambas variables mediante `scanf`
    - Calcular x^n a base de multiplicar x tantas veces como indique n.
    - Mostrar el resultado utilizando un formato que sólo muestre 2 cifras en la parte decimal.

16. Realizar un programa en C que reconozca si una palabra es un palíndromo. Una palabra es un palíndromo si se lee igual de izquierda a derecha que de derecha a izquierda. Para reconocer si una palabra es o no un palíndromo utilizar una función. Para simplificar el problema, escribir el texto todo en mayúsculas y sin acentos.

```
int Palindromo (char palabra[]);
```

    - Devuelve 1: si la cadena "palabra" es palíndromo. Se ignorarán los espacios en blanco. Por ejemplo: "DABALE ARROZ A LA ZORRA EL ABAD" o "SALTA LENIN EL ATLAS" son palíndromos.
    - Devuelve 0: si no lo es.

17. Realiza el código de la función LimpiarCaracter() que recibe tres argumentos: una cadena de caracteres y dos caracteres c1 y c2. La función se encargará de buscar en la cadena todas las ocurrencias del primero de los caracteres (c1) y sustituirlo por el segundo (c2) en todos los casos en los que se haya encontrado. Además, la función debe devolver:
    - el valor 0 si no ha encontrado ninguna ocurrencia de c1
    - el número de veces que lo ha encontrado, caso de que así haya sido.

18. Escribir un programa que pida una fecha en dos variables: día y mes.
    - Primero se debe pedir el mes, validando que se encuentra entre 1 (enero) y 12 (diciembre).
    - Luego se pedirá el día, validando que se encuentra entre 1 y 31, entre 1 y 30, o entre 1 y 28, dependiendo del valor del mes.
    - Finalmente escribir por pantalla la fecha leída.

    Notas:
    - Se considera que el año no es bisiesto.
    - Validar implica comprobar si el valor introducido es válido y en caso de que no lo sea, solicite de nuevo otro valor.

19. Escribir una función EliminarVocales(char cadena[]) que reciba una cadena del programa principal y la transforme en otra (generalmente más corta) donde no estén las vocales. Si recibe, por ejemplo, la cadena "Ordenador", la debe transformar en la cadena "rdndr".

20. Escribir un programa que solicite al usuario la cantidad de números n que va a introducir, debiendo validar que n sea un valor positivo. A continuación, vaya solicitando tantos números enteros como haya indicado el usuario con n, y muestre por pantalla los números que sean múltiplos de 3. Finalmente, deberá mostrar al usuario la media de los valores que no son múltiplos de 3. NO PUEDEN USARSE ARRAYS EN LA SOLUCION.

21. Escribe la función void InvertirCadena (char *cadena) que invierte el orden de los caracteres de la cadena especificada en el argumento y devuelve un puntero a la cadena resultante.

22. Un profesor de colegio necesita imprimir una tabla de los divisores de los números enteros para los alumnos de su clase. Completar el programa en C que la proporciona. Para crear la tabla, inicialmente hay que pedir al usuario un número entero positivo n. A continuación el programa llamará repetidamente a una función Escribir_Divisores() para obtener los divisores de cada número desde 1 hasta n.

    Se pide:
    - escribir en el main la **llamada a la función**
    - escribir **la función Escribir_Divisores()**

    Un ejemplo de la ejecución de este programa puede ser lo siguiente:

```
Introduzca un numero entero positivo:15
Los divisores de numeros de 1 hasta 15 son:
-----------------------------------------------------------
1 --> 1,
2 --> 1, 2,
3 --> 1, 3,
4 --> 1, 2, 4,
5 --> 1, 5,
6 --> 1, 2, 3, 6,
7 --> 1, 7,
8 --> 1, 2, 4, 8,
9 --> 1, 3, 9,
10 --> 1, 2, 5, 10,
11 --> 1, 11,
12 --> 1, 2, 3, 4, 6, 12,
13 --> 1, 13,
14 --> 1, 2, 7, 14,
15 --> 1, 3, 5, 15
```

23. Escribir un programa que encuentre una matriz de números reales simétrica. Para ello, una función entera con entrada la matriz determinará si ésta es simétrica. En otra función se generará la matriz con números aleatorios de 1 a 19. Utilizar la aritmética de punteros en la primera función y en la segunda indexación.

24. Escribir un programa completo que solicite un número entero positivo al usuario (debe validarlo) y calcule la suma de las cifras de dicho número. Finalmente, el programa debe mostrar un mensaje al usuario con el resultado de la suma.

    Ejemplo 1 de ejecución:

    ```
    Introduzca un numero entero positivo: -12

    El numero debe ser positivo.

    Introduzca un numero entero positivo: 3791

    La suma de las cifras del numero 3791 es 20.
    ```

    Ejemplo 2 de ejecución:

    ```
    Introduzca un numero entero positivo: 229

    La suma de las cifras del numero 229 es 13.
    ```

25. En el siguiente código se accede a los elementos de una matriz. Modificar el código para acceder a los mismos elementos con aritmética de punteros.

```
#include <stdio.h>

// Declaración de constantes
#define NFILAS 4
#define NFILAS 5

// Declaración de variables
int  fila, columna;
double matrizt[NFILAS][ NFILAS];

// Programa principal. Solo bucle for

int main()
{
    for (fila= 0; fila <N ; fila ++){
        for (columna = 0; columna <M; columna ++)
       printf("%lf  ", matriz[fila][ columna]);
    }
}
```

26. Desarrollar un conjunto de rutinas de pila para guardar y recuperar valores enteros. La pila es una lista que utiliza un acceso a tipo primero en entrar, último en salir. Para crear la pila se necesitan dos funciones: push() y pop(). La función push() coloca (guarda) valores en la pila y pop() los saca (recupera). El programa debe guardar en la pila los valores que introduzca el usuario. Si se introduce un 0, se recupera un valor de la pila. Para finalizar el programa, se tiene que introducir un -1.

27. Realice un programa que almacene en una variable (punt_array) la dirección de memoria de un array de tipo char (llamado array_char) inicializado con el valor ['X','Y','Z']. Ahora crear una función imprime array que imprime las 3 posiciones del array, pasar a esta función punt_array en una llamada y en una segunda llamada enviar array_char por referencia.