

1. Escribir la cabecera o prototipo de la función para cada una de las situaciones que se describen:

- Calcular las raíces de una ecuación de segundo grado
- Determinar si un número tiene como factores primos al 3 y al 11
- Calcular cuantos primos hay en un intervalo
- Sacar por la salida estándar los primos que hay en un intervalo

2. Describir la salida de los siguientes programas:

```
#include <stdio.h>

void P(int i, int j) {
    i = i+1;
    j = j+3;
    printf("%d %d\n", i, j);
}
```

```
int main() {
    int a, b;

    a = 2;
    b = 7;
    P(a,b);
    printf("%d %d\n", a, b);
    return 0;
}
```

```
#include <stdio.h>

void P(int *i, int *j) {
    (*i) = (*i)+1;
    (*j) = (*j)+3;
    printf("%d %d\n", (*i), (*j));
}
```

```
int main() {
    int a, b;

    a = 2;
    b = 7;
    P(&a,&b);
    printf("%d %d\n", a, b);
    return 0;
}
```

3. Encuentre los errores de las siguientes funciones:

```
int sum (int n) {
    if (0 == n)
        return 0;
    else
        n = n + 5;
}
```

```
void f(float a) {
    float a;

    printf("%f", a);
}
```

```
void producto (int a) {
    return a*a;
}
```

4. Realizar un programa para que escriba todos los números primos entre 1 y N, siendo N un número introducido por el usuario.

5. Escribir una función en C para aceptar mensajes de confirmación. La función debe imprimir en la pantalla el mensaje **¿Confirmar (S/N)?** y sólo aceptará las pulsaciones de las teclas **S** y **N** (en mayúscula o minúscula). La función devolverá **false** (0) si se ha pulsado **N** o bien **true** (1) si se ha pulsado **S**. Hacer un main que muestre como se usa la función.

6. Implemente una función en C con un argumento de tipo carácter que:

- si el argumento es una letra en mayúscula, devuelve su correspondiente letra en minúscula,
- si el argumento no es una letra mayúscula, devuelve el mismo argumento.

Hacer un pequeño main que ilustre el uso de la función.

7. Escribir en C la función `int MCD(int a, int b)` que devuelve el máximo común divisor de dos números enteros. Hacer un main para ilustrar su uso.
8. Escribir un programa modular que presente un menú para calcular la potencia, factorial y combinatorio de ciertos valores leídos desde el teclado y muestre en pantalla el resultado de la operación. Hacer primero la descomposición modular.
9. Escribir una función que nos diga si un año dado (como un entero) es o no bisiesto. Los años bisiestos son aquellos que bien son múltiplo de 4 pero no de 100, o bien son múltiplo de 400. Además, hay que tener en cuenta que este calendario empezó a aplicarse a partir de 1582.
10. Escribir una función que, dados tres enteros representando el día, mes y año, nos diga si corresponden a una fecha correcta.
11. Escribir en C la función `int MCM(int a, int b)` que devuelve el mínimo común múltiplo de dos números enteros.

12. Mostrar qué escriben los siguientes programas, para las entradas 1, 3 y 7:

```
#include <stdio.h>
```

```
void sumar(int *x, int *a,
           int *z, int sum) {
    scanf("%d", x);
    scanf("%d", a);
    scanf("%d", z);
    sum = (*x) + (*a) + (*z);
}
```

```
int main() {
    int a, b, c, sum=0;

    sumar(&a,&b,&c,sum);
    printf("%d + %d + %d = %d\n", a, b, c, sum);
    return 0;
}
```

```
#include <stdio.h>
```

```
void sumar(int *x, int *a,
           int *z, int *sum) {
    scanf("%d", x);
    scanf("%d", a);
    scanf("%d", z);
    (*sum) = (*x) + (*a) + (*z);
}
```

```
int main() {
    int a, b, c, sum=0;

    sumar(&a,&b,&c,&sum);
    printf("%d + %d + %d = %d\n", a, b, c, sum);
    return 0;
}
```

13. Hacer una función reciba un número entero y que devuelva el número de cifras que este número entero tiene. Hacer un programa que muestre el uso de la función.
14. Escribir una función que calcule el producto de la primera y de la última cifra de un número entero recibido como argumento. Hacer un programa que muestre el uso de esta función.
15. Realizar un pequeño programa en C que cuenta el número de caracteres, palabras y líneas de un fichero de texto (que lee de stdin haciendo uso de redirección de entrada).
16. La fuerza de atracción entre dos masas, m_1 y m_2 separadas por una distancia d , está dada por la fórmula: $F = \frac{G \cdot m_1 \cdot m_2}{d^2}$ Donde G es la constante de gravitación universal $G = 6,67 \cdot 10^{-11}$ Escriba un programa que lea la masa de dos cuerpos y la distancia entre ellos y a continuación obtenga la fuerza gravitacional entre ellas.
17. Escribe un programa que lea tres números reales que representan una hora en formato horas, minutos y segundos, le sume un segundo y saque el resultado en formato HH:MM:SS.
18. Escribe un programa que lea un número natural que representa un número de segundos y lo descomponga en horas, minutos y segundos.
19. Sabiendo que las distancias de los planetas del sistema solar en millones de kilómetros al sol son:

1	Mercurio	59
2	venus	108
3	Tierra	150
4	Marte	228
5	Júpiter	750
6	Saturno	1431
7	Urano	2877
8	Neptuno	4509
9	Plutón	5916

Hacer un programa que muestre un menú de planetas para que el usuario seleccione un planeta y el programa nos diga su distancia al sol. (Utilizar la sentencia switch).

20. Hacer una función que reciba cuatro números enteros que son el numerador y denominador de dos fracciones sume estas dos fracciones y devuelva dos números enteros que sera el numerador y denominador de la fracción suma. Hacer también un programa principal para ilustrar el uso de la función.
21. Realizar un programa que pida números enteros hasta que se introduzca el cero. El programa debe sumar todos los números pares y los números impares para al finalizar mostrar estas sumas.
22. Hacer una función para calcular el cociente y el resto de una división entera. La función recibirá los dos números dividendo y divisor y devolverá los dos números cociente y resto. No se puede utilizar la función módulo y la división entera hay que hacerlo por restas sucesivas. Hacer un main que ilustre como se usa la función.
23. Realizar un programa que permita calcular el área de un cuadrado, un círculo o un triángulo equilátero. El programa ha de presentar un menú pedir los datos necesarios y hacer el cálculo.
24. Hacer un programa que calcule el área y el perímetro de un rectángulo dada la base y la altura del rectángulo.
25. Escribir un programa (con las funciones que se estime pertinentes) que escriba todos los números perfectos menores que 1000. Se dice que un número es perfecto cuando es igual a la suma de todos sus divisores (incluyendo el 1 y excluyendo el propio número).
26. El método de Newton para hallar la raíz cuadrada de un número real positivo X tiene gran interés ya que utiliza sólo sumas multiplicaciones y divisiones (algunas de ellas divisiones por 2 que son inmediatas para la máquina). El método consiste en lo siguiente: para calcular la raíz cuadrada de un número X tal que el cuadrado de la solución difiera de X menos de un cierto error E , comenzamos con la aproximación $a = X/2$. Si $|(a * a) - X| < E$ paramos los cálculos y el resultado es a . Si no, reemplazamos a con la siguiente aproximación definida por $(a + X/a)/2$. Entonces comprobamos si esta aproximación es lo suficientemente buena de la misma manera que antes lo hicimos. Si lo es, el cálculo finaliza y si no prosigue iterativamente, estando garantizado que la aproximación converge. Escriba una función que implemente este método dados X y el error E aceptable, y un programa que la utilice.
27. Hacer una implementación recursiva para la función $factorial(X)$. La definición recursiva de esta función dice que el $factorial(0)$ es 1 y para cualquier otro X el $factorial(X)$ es $X * factorial(X - 1)$.
28. Crear un programa que encuentre el máximo común divisor de dos números usando el algoritmo recursivo de Euclides : Dado dos números enteros positivos m y n , tal que $m > n$, para encontrar su máximo común divisor, es decir, el mayor entero positivo que divide a ambos:
 - Dividir m por n para obtener el resto r ($0 \leq r < n$).
 - Si $r = 0$, el MCD es n .
 - Si no, el máximo común divisor es $MCD(n,r)$.