



---

**FUNDAMENTOS DE INFORMÁTICA**  
**PROBLEMAS DE PROGRAMACIÓN**  
**TEMA 2. LENGUAJE C: CONCEPTOS BÁSICOS Y PROGRAMACIÓN ELEMENTAL**

---

1. Codificar en C todos los algoritmos propuestos en la hoja de ejercicios nº 1.
2. Indicar el tipo y el significado de las siguientes constantes
  - a. 'a'
  - b. 27
  - c. 0xAA
  - d. 0512
  - e. 0x6C4
  - f. 99.56
  - g. -2e30
3. Teniendo en cuenta que en un programa las variables **x** e **y** están declaradas del siguiente modo: **int y,x;**, obtener el valor de **y** en las siguientes expresiones en C. En todos los casos, suponer que el valor inicial de las variables es **x =5, y =5;**
  - a.  $y^* = x + 1;$
  - b.  $y /= (5+x);$
  - c.  $y = ++x - 2;$
  - d.  $y = x \% (x+2);$
  - e.  $y = (\text{float}) x/2;$
4. Escribir el resultado de las siguientes expresiones, donde **x** e **y** son **variables enteras**: (Para cada expresión el valor de **x** e **y** inicial es el que tomaban en la anterior expresión).
  - a.  $x = 24 \% 5;$
  - b.  $x = (7 >= 5 \ \&\& \ 27 == 8)$
  - c.  $x = (45 <= 7 \ || \ !(5 >= 7))$
  - d.  $x = (2+3)^*6;$
  - e.  $x = 2+3^*6;$
  - f.  $x = (12+6)/2^*3;$
  - g.  $y = x = (2+3)/4;$
  - h.  $2+3=x;$
  - i.  $y = x = 1;$
  - j.  $y = 3+2^*(x=7/2);$
  - k.  $x=(2+3)^*10.5;$
  - l.  $x = 3/5^*22.0;$
  - m.  $x = (3==5)^*7;$
  - n.  $x = 3==5^*7;$
  - o.  $x = 2^*(7<=5 \ \&\& \ 2>=3);$
  - p.  $x = 2^*(7<=5 \ || \ 2>=3);$
  - q.  $x = 3!=7;$
  - r.  $x = 'a' > 'c';$



- s.  $y = 6 + x++;$
- t.  $y /= x;$
- u.  $y += ++x;$
- v.  $x = (y == x);$

5. Escribir las siguientes condiciones de entrada de una sentencia *if*, siendo *x* e *y* **variables enteras** y *c* una **variable char**.
- a. *x* debe ser menor a 5 o mayor a 10;
  - b. *x* debe ser menor a 5 o mayor a 10 y no puede ser negativo;
  - c. *x* debe ser negativo e *y* debe estar comprendido entre 2 y 5;
  - d. *x* debe ser distinto de 0, e *y* debe ser mayor a 10;
  - e. *c* debe ser una letra mayúscula que no sea una vocal;
  - f. *x* debe ser menor a 3 o mayor a 5, o bien *y* es igual a 5;
6. Dado el programa siguiente, se pide:
- a. Realizar un seguimiento del programa y escribir la salida que se obtendría al ejecutarlo.
  - b. Transformar los bucles *while* a bucles *do – while* y *for*.

```
#include <stdio.h>
void main(void)
{
    int a,b;

    a = 10;
    while (a != 0)
    {
        if (a == 3)
            break;
        else
            b = a + 1;
        while (b > 0)
        {
            b = b - 1;
            if (b == 5)
                continue;
            else
                printf("%d", b);
        }
        printf("\n");
        a = a-1;
    }
}
```

7. Dado el programa siguiente, se pide:
- a. Realizar un seguimiento del programa y escribir la salida que se obtendría al ejecutarlo.
  - b. Transformar el bucle *for* en bucle *while* y *do - while*



```
#include <stdio.h>
void main(void)
{
    int f,c;

    for (f=0; f<4; f++)
    {
        for (c=0; c<5; c++)
            printf(" %d,%d ", f,c);
        printf("\n");
    }
}
```

8. Escribir un programa en C que lea un número entero e imprima todos sus divisores.
9. Modificar el programa anterior para que se calculen los divisores de todos los números enteros que vaya introduciendo el usuario hasta que introduzca un valor negativo. La ejecución debe ser del siguiente modo: Se pide un número y se muestran por pantalla sus divisores, se pide otro número y se muestran sus divisores. Esta ejecución continua hasta que se introduce un número negativo.
10. ¿Cuál es el número mínimo de veces que se ejecuta un bucle *do – while* ? Compararlo con un bucle *while* y explicar los detalles en los que se diferencian.
11. Se desea escribir por pantalla la suma de los números 2, 5, 8, 11, 14, ... (números incrementados de tres en tres), hasta *n*, que es una variable entera leída por teclado. Escribir un bucle que calcule esta suma de tres formas:
  - c. Utilizando un bucle *while*
  - d. Utilizando un bucle *do – while*
  - e. Utilizando un bucle *for*
12. Escribir un programa que lea una cantidad *n* de números enteros por teclado (el valor *n* lo debe pedir al principio al usuario), y nos devuelva como resultado la suma de todos los valores introducidos y su valor medio. Asimismo, debe indicar cuantos de los números introducidos son múltiplos de 3.
13. Escribir un programa que lea una cantidad *n* de números enteros por teclado (el valor *n* lo debe pedir al principio al usuario), y nos devuelva como resultado la suma de todos los valores introducidos y su valor medio. Asimismo, debe indicar cuantos de los números introducidos son múltiplos de 3. El programa debe finalizar en cualquier momento si introducimos el número 100.
14. Escribir un programa que lea números enteros por teclado hasta que se introduzca un valor negativo, y nos devuelva como resultado, la suma de todos los números introducidos múltiplos de 3 y el producto de todos los números introducidos que son múltiplos de 2.



15. Escribir un programa que lea números enteros por teclado hasta que se introduzca un valor negativo, y nos devuelva como resultado, la suma de todos los números introducidos excepto aquellos que sean múltiplos de 3 o de 5;
16. Escribir un programa en C que calcule la suma de los  $n$  primeros números pares y el producto de los  $n$  primeros números impares simultáneamente, donde  $n$  es un entero que se lee por teclado.
17. Escribir un programa en C que dados los coeficientes de un polinomio de segundo grado  $ax^2+bx+c$ , que se leerán por teclado, obtenga todas sus raíces.
18. Escribir una instrucción *switch* que examine el valor de una variable entera llamada *indicador* y escriba uno de los siguientes mensajes dependiendo de su valor:
  - f. CALOR, si *indicador* tiene el valor 1.
  - g. TEMPLADO, si *indicador* tiene el valor 2.
  - h. FRIO, si *indicador* tiene el valor 3.
  - i. FUERA DE RANGO, si *indicador* tiene el valor 4.
  - j. ERROR, si *indicador* tiene cualquier otro valor.
19. Escribir la salida de los siguientes programas en C

```
#include <stdio.h>

void main(void)
{
    int i=0, x=0;

    while (i<20) {
        if (i%5 == 0) {
            x+=i;
            printf("%d ", x);
        }
        ++i;
    }
    printf("\nx = %d", x);
}
```

```
#include <stdio.h>

void main(void)
{
    int i=0, x=0;

    do{
        if (i%5 == 0) {
            x++;
            printf("%d ", x);
        }
        ++i;
    }while(i<7);
    printf("\nx = %d\n", x);
}
```

```
#include <stdio.h>

void main(void)
{
    int i=0, x=0;

    for(i=1; i<10; i++) {
        if (i%2 == 1)
            x+=i;
        else
            x--;
        printf("%d ", x);
    }
    printf("\nx = %d\n", x);
}
```



```
#include <stdio.h>

void main(void)
{
    int i,j,k,x=0;

    for(i=0; i<5; i++)
        for (j=0; j<i; j++) {
            switch(i+j-1) {
                case -1:
                case 0:
                    x += 1;
                    break;
                case 1:
                case 2:
                case 3:
                    x += 2;
                    break;
                default:
                    x +=3;
            }
            printf("%d ",x);
        }
    printf("\nx = %d\n", x);
}
```

```
void main(void)
{
    int i,j,n;
    n=8;
    i = n;
    do
    {
        j = i;
        do
        {
            printf(" %d", j);
            j = j-1;
        }while(j>0);
        printf("\n");
        i = i-1;
    }while(i>0);
}
```

20. El recibo de la electricidad se elabora de la siguiente forma:

- 300 ptas. de gastos fijos.
- 50 ptas/Kwh para los primeros 100 Kwh.
- 35 ptas/Kwh para los siguientes 150 Kwh.
- 25 ptas/Kwh para el resto.

Escribir un programa que lea del teclado los dos últimos valores del contador, y calcule e imprima en pantalla el importe total a pagar.

21. Escribir un programa que lea los valores de 3 resistencias eléctricas (en Ohmios) conectadas en paralelo y muestre en pantalla el valor equivalente de las mismas. El valor equivalente se calcula como  $1/(1/R1 + 1/R2 + 1/R3)$ . Modificar el programa anterior para:
- a. Aplicar la fórmula a cualquier cantidad de resistencias. En primer lugar, el programa pedirá el número de resistencias a conectar en paralelo. Después pedirá uno a uno los valores de todas las resistencias. Por último, escribirá el resultado equivalente.
  - b. Que muestre también el resultado si todas las resistencias se conectan en serie. Recordar que en este caso la resistencia equivalente es la suma de todas las resistencias.
22. Escribir un programa que lea un número natural  $n$  y dibuje un triángulo de asteriscos (\*) con base y altura  $n$ .
23. Escribir un programa para encontrar el menor número tal que: dividido entre 2 da de resto 1, dividido entre 3 da de resto 2, dividido entre 4 da de resto 3, dividido entre 5



da de resto 4, dividido entre 6 da de resto 5, dividido entre 7 da de resto 6, dividido entre 8 da de resto 7 y dividido entre 9 da de resto 8. ¿Qué ocurriría en la ejecución del programa si no existe un número que cumpla estas condiciones?

24. El N.I.F. (Número de Identificación Fiscal) es un número único para cada persona formado por el D.N.I. y una letra añadida. Esta letra depende directamente del número del D.N.I., y se introdujo para poder detectar posibles errores al teclear o escribir un determinado D.N.I. La letra es simplemente un mecanismo de *control de errores*.

El algoritmo para calcular la letra a partir del número del D.N.I. es el siguiente. Se calcula el resto de dividir el D.N.I. entre 23. Este resto estará entre 0 y 22 y a cada uno de esos posibles restos se le asigna respectivamente las siguientes letras: T, R, W, A, G, M, Y, U, P, D, X, B, N, J, Z, S, Q, V, H, L, C, K y E. Se ha escogido el número 23 por ser el número primo más grande que es menor que el número de letras en castellano.

Se pide:

- Escribir un programa que lea un número de D.N.I. y escriba la letra del N.I.F. que le corresponde.
- Modificar el programa anterior para que dado un N.I.F. indique si es correcto o no. En caso de ser incorrecto deberá indicar la letra que le corresponde.

25. Identificar y corregir los errores del siguiente programa en C:

```
#include <stdio.h>

void main(void)
{
    float h, d, t, n;

    printf("Introduce h: ");
    scanf("%f", &h);
    printf("Introduce d: ");
    scanf("%f", &d);
    h*d = t;
    n = t-j;
    printf("El resultado es: %12.5f, total);
    printf("El valor es: %12.5f", N);
}
```