



Problemas del tema: Sentencias de Control: Segunda Parte (más....)

1. ¿Cuál es el efecto de las siguientes instrucciones? (revisar cuidadosamente)

```
if (a > 2);  
b = 2;  
c = 3;
```

2. Los siguientes programas pretenden mostrar por pantalla la suma total de los enteros entre 1 y 10 (ambos incluidos), pero en todos ellos hay algún error que provoca que la salida sea incorrecta. Corrige los errores para que los programas funcionen correctamente.

a) `#include <stdio.h>`

```
  
int main(void) {  
    int i = 1;  
    int sum = 0;  
  
    while (i < 11) {  
        sum = sum + i;  
        ++i;  
        printf("%d\n", sum);  
    }  
}
```

b) `#include <stdio.h>`

```
  
int main(void) {  
    int i;  
    int sum = 0;  
  
    for (i = 0; i >= 10; ++i)  
        sum = sum + i;  
    printf("%d\n", sum);  
}
```

c) `#include <stdio.h>`

```
  
int main(void) {
```

```
int i = 0;
int sum = 0;

while (i <= 11)
    sum = sum + i++;
printf("%d\n", sum);
}
```

3. El siguiente código C fue escrito para encontrar el mínimo de un conjunto de enteros positivos que el usuario introduce por teclado. El conjunto de números termina cuando el usuario introduce el valor -1 . Una vez que se han introducido todos los números el programa muestra en la pantalla el valor del número más pequeño. Sin embargo, el código contiene un error; encuéntralo y arregla el programa.

```
#include <stdio.h>

int main(void) {
    int Minimo = 0;
    int Siguiente;

    /* Lee el primer valor */
    scanf("%d", &Siguiente);

    while (Siguiente != -1) { /* Mientras no se lea el valor -1 */
        if (Siguiente < Minimo)
            Minimo = Siguiente;
        scanf("%d", &Siguiente); /* Se lee el siguiente valor */
    }
    printf("Num. mas pequeño ->%d\n", Minimo);
}
```

4. ¿Cuál es la salida de los siguientes códigos?:

a)

```
x=10;
while (x > 10)
    x = x - 3;
printf("x =%d\n", x);
```

b)

```
x=10;
while (x > 10)
    x = x + 4;
printf("x =%d\n", x);
```

c)

```
x=10;
do
    x = x - 3;
while (x <= 0);
printf("x =%d\n", x);
```

```

d)      for (i = -3; i <= 11; i++)
          printf("2 * i =%d\n", (2 * i));

e)      for (i = 10; i >= 1; i--)
          printf("i =%d\n", i);

f)      for (j = 10; j >= 1; j--)
          for (i = 1; i <= 10; i++)
              printf("%d * %d =%d\n", i, j, (i * j));
    
```

5. ¿Cuál es la salida de los siguientes códigos cuando $x = 0$ y cuando $x = 1$?:

```

a) if (x = 0)
    printf("x igual a 0\n");
    else
    printf("x no es igual a 0\n");

b) if (x == 0)
    printf("x igual a 0\n");
    else
    printf("x no es igual a 0\n");

c) if (x == 0)
    printf("A\n");
    else
    if (x != 1)
    printf("B\n");
    else
    if (x < 1)
    printf("C\n");
    else
    if (x)
    printf("D\n");
    
```

```
d) int x;
    int y;

    switch(x) {
        case 0: y = 3;
        case 1: y = 4;
                break;
        default: y = 5;
                break;
    }
    printf("y = %d\n", y);
```

e) ¿Cuál es la salida del apartado anterior si x es diferente de 0 y de 1?

6. ¿Se puede convertir esta sentencia *if-else* a un *switch*?. Si se puede, hazlo. Si no, explica por qué.

```
if (x == 0)
    y = 3;
else
    if (x == 1)
        y = 4;
    else
        if (x == 2)
            y = 5;
        else
            if (x == y)
                y = 6;
            else
                y = 7;
```

7. ¿Cuál es la salida del siguiente código?

```
int r = 0;
int s = 0;
int w = 5;
int suma = 0;

for (r = 1; r <= w; r++)
    for (s = r; s <= w; s++)
        suma = suma + s;

printf("suma = %d\n", suma);
```

8. ¿Cuál es la salida de los siguientes segmentos de código?.

- a)
- ```
int x = 20;
int y = 10;

while ((x > 10) && (y < 15)) {
 y = y + 1;
 x = x - 1;
 printf("*");
}
```
- b)
- ```
int x;

for (x = 10; x; x = x - 1)
    printf("*");
```
- c)
- ```
int x;

for (x = 0; x < 10; x = x + 1) {
 if (x % 2)
 printf("*");
}
```
- d)
- ```
int x = 0;
int i;

while(x > 10) {
    for (i = 0; i < x; i = i + 1)
        printf("*");
    x = x + 1;
}
```

9. Escribir un programa que lea tres números enteros y asigne el mayor a la variable *max*.
10. Escribir un programa en C que lea un entero de teclado y muestre un 1 por pantalla si el número es divisible por 3 y un 0 en otro caso.
11. Escribir un programa que lea un número y muestre un mensaje por pantalla que indique si el número leído es par o impar.
12. Escribir un programa que lea dos números de teclado, *num1* y *num2*, y muestre el producto de los dos si *num1* es menor que *num2* y la división de *num1* entre *num2* si no lo es.
13. Escribir un programa que determine si un año es bisiesto. Un año es bisiesto si es múltiplo de 4, excepto los múltiplos de 100, que no son bisiestos, salvo que a su vez también sean múltiplos de 400.
14. Escribir un programa que lea dos números por teclado, *x* e *y*. Si *x* es un 1 debe mostrar el valor de *y* por pantalla. Si *x* es un 2, debe mostrar el producto de los dos números. Si *x* es un 3, hay que mostrar la suma de los dos. Si es cualquier otro valor hay que mostrar el resultado de la operación $y - x$.

15. Diseñar un algoritmo que, tras leer tres números, compruebe si la suma de cualquier pareja de ellos es igual al tercer número.
16. En una empresa los salarios se van a incrementar según su contrato actual. Escribir un programa que solicite el salario actual del empleado y calcule y visualice el nuevo salario según la siguiente tabla:

Contrato	Aumento
0 a 1000 euros	20 %
1001 a 2000 euros	10 %
2001 a 3000 euros	5 %
más de 3000 euros	0 %

17. Imprimir la tabla de multiplicar de un número leído por teclado.
18. Calcular el factorial de un número leído por teclado.
19. Escribir un programa que calcule la suma: $1 + 2 + 3 + \dots + 100$.
20. Escribir un programa que escriba todos los números pares comprendidos entre 1 y 50.
21. Escribir un programa que muestre en pantalla los múltiplos de 3 entre 3 y 99.
22. Escribir un programa para visualizar por pantalla las potencias de 2 comprendidas entre 1 y 10000.
23. Escribir un programa que lea números negativos de teclado y cuente el número de valores leídos. La lista de números terminará cuando se introduzca un cero.
24. Escribir un programa que lea números de teclado hasta que se introduzca un 0. El algoritmo debe sumar los valores positivos y calcular el producto de los números negativos.
25. Escribir un programa que calcule el producto de dos valores enteros leídos de teclado, utilizando la suma como instrucción básica.
26. Escribir un programa que lea dos números enteros y calcule su cociente y su resto, utilizando la resta como operación básica.
27. Escribir un programa que pida dos números y calcule su Máximo Común Divisor (mcd), utilizando la siguiente fórmula:

$$mcd(a, b) = \begin{cases} a & \text{si } a = b \\ mcd(a, b - a) & \text{si } a < b \\ mcd(a - b, b) & \text{si } a > b \end{cases}$$

28. Escribir un programa que determine si un número entero, Y , es un cuadrado perfecto; es decir, si existe un número entero X , tal que $X^2 = Y$.
29. Escribir un programa que calcule la suma: $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$, donde n es un valor introducido por teclado.
30. Escribir un programa que calcule la suma: $\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \dots + \frac{n}{2^n}$, donde n es un valor introducido por teclado.
31. Escribir un programa que, dado un número, determine la suma de sus dígitos.

32. Un número perfecto es un entero positivo que es igual a la suma de sus divisores propios. Un divisor propio es un entero positivo diferente del número en sí mismo que divide al número de forma exacta. Escribe un algoritmo que calcule si un número es perfecto. Ejemplos:
 - Divisores propios de $6 = 1, 2, 3 \rightarrow 1 + 2 + 3 = 6 \rightarrow 6$ es un número perfecto
 - Divisores propios de $8 = 1, 2, 4 \rightarrow 1 + 2 + 4 = 7 < 8 \rightarrow 8$ no es un número perfecto
33. Se ha comprobado que si a un número natural se le somete al siguiente proceso:
 - a) Si el número es par se divide por 2
 - b) Si el número es impar se multiplica por 3 y se le suma 1
 - c) Si el número es igual a 1 se para el proceso, si no, volver al primer puntoTodos los números naturales acaban tomando el valor 1 después de un cierto número de iteraciones del proceso. Escribir un algoritmo que calcule el número de iteraciones para llegar a 1 para un entero dado.
34. Escribir un programa que acepte 5 números por teclado y calcule su media.
35. Modifica el programa anterior para que se permita leer 100 números.
36. Escribir un programa que lea un vector de N elementos y sustituya los valores negativos por el máximo del vector.
37. Escribir un programa que lea de teclado las temperaturas medias alcanzadas cada día de un año en Santa Cruz de Tenerife.
38. Escribir un programa que lea 10 números de teclado, calcule el doble de cada número y muestre por pantalla el resultado obtenido.
39. Escribir un programa que lea de teclado dos vectores de 20 elementos cada uno y obtenga el vector suma, cuyo valor en la posición i se calcula como la suma de los valores que se encuentran en la posición i de los dos vectores.
40. Escribir un programa que lea N números flotantes de teclado y después pida un valor x . El programa debe sustituir todos los valores menores de x por 0.
41. Escribir un programa que lea 2 vectores de N enteros y genere un nuevo vector, donde el valor de la posición i se calculará como el valor máximo de las posiciones i de los dos vectores.
42. Escribir un programa que lea un vector de N elementos y un valor k . Debe calcular el resultado de multiplicar k por cada elemento del vector y almacenarlo en un nuevo vector. Para finalizar, debe mostrar por pantalla el nuevo vector calculado.
43. Escribir un programa que lea una cadena de caracteres de teclado y calcule la longitud de la cadena leída, sin utilizar la función *strlen*.
44. Escribir un programa que lea cadenas de caracteres de teclado y cuente el número de cadenas leídas. El programa finalizará cuando se lea la cadena *adios*. La última cadena también debe contarse.
45. Escribir un programa que lea dos cadenas de caracteres y las compare, mostrando por pantalla un mensaje indicando si son iguales o no. En el caso de que no lo sean, se debe mostrar también la cadena más pequeña.

46. Escribir un programa que lea de teclado una matriz de $N \times M$, calcule el doble de cada valor y los muestre por pantalla.
47. Escribir un programa que lea una matriz de $N \times M$ y calcule la media de todos los valores.
48. Escribir un programa que lea una matriz de $N \times M$ números flotantes y calcule el valor más alto leído.
49. Escribir un programa que lea una matriz de 100×100 , pida un número k por teclado y modifique la matriz, de forma que todos los elementos menores que k se sustituyan por 0.
50. Escribir un programa que lea de teclado una matriz de tamaño $N \times M$ y calcule la suma de los elementos por columnas, almacenando los resultados en un vector.
51. Escribir un programa que lea de teclado una matriz de 3×3 elementos y calcule la suma de su diagonal principal.
52. Escribir un programa que lea de teclado los datos de una matriz de tamaño $N \times M$ y cuente el número de veces que aparece el número 0 y lo almacene en la variable `num_ceros`. A continuación, debe sustituir todos los elementos donde haya un 0 por el valor de `num_ceros`.
53. Escribir un programa que pida por teclado la altura, peso y sexo de 10 personas y muestre por pantalla la media de las alturas y de los pesos y el número de hombres y de mujeres.
54. Escribir un programa que pida por teclado la marca, modelo y número de Km. recorridos de 50 coches y calcule el número medio de Km. recorridos y la marca y modelo del coche que tiene más Km. y del que tiene menos.
55. Considerando la estructura para representar el horario de un tren, vista en la hoja de problemas del tema *Variables y Operadores*, escribir un programa que lea por teclado los datos de 10 trenes y los almacene en un array.
56. Considerando la estructura para representar libros y revistas de una biblioteca, vista en la hoja de problemas del tema *Variables y Operadores*, escribir un programa que lea 10 libros o revistas y las almacene en un array.
57. En una secretaría de un Centro Universitario se concede cita previa a 20 alumnos al día para realizar su matrícula. Para cada alumno se almacena su nombre, apellidos y dni. Escribir un programa que presente un menú que permita leer de teclado los datos de un nuevo alumno, o mostrar por pantalla todos los alumnos con cita previa.