

FUNDAMENTOS DE INFORMÁTICA

TEMA 4: ESTRUCTURAS DE CONTROL

EJERCICIOS

Nivel 1 : Ejercicios Fáciles

1. Indica cual es el valor numérico y lógico (verdadero o falso) de las siguientes expresiones, dado que la variable a ha sido inicializada con el valor 0, b con el valor 10 y c con el valor -6.

a	!a	!!b	a>0
!(a>b)	b>=a+c	a>b&& c<5	a++
++a	!(a>b) !(a!=b)	(a=b)	!(a=!b)
!(!(a&&!b) !(!c !b&&!a))			

2. Haciendo uso del operador condicional, escriba expresiones para clasificar un carácter basándose en las siguientes condiciones:

- (a) Verdadero si un carácter es un dígito decimal en el rango 0-9 y falso en otro caso.
 (b) Verdadero si el carácter es un dígito hexadecimal (0-9 ó 'A'-'F') y falso en otro caso.

3. Ordena el siguiente trozo de programa para que tenga una indentación (o sangrado) correcta. Observa a qué if pertenece cada caso.

```
if(b!=0) if (b+3>a) sentencia0;
else if(a+b<c) sentencia1; else if (b<10) sentencia2;
else {sentencia3; if(a-b>x) sentencia4; } else sentencia5;
```

4. Considera el siguiente fragmento de programa:

```
if(a>=0)
    if(a<=5)
        puts("a está en el rango entre 0 y 5");
    else
        puts("a debe ser menor que cero");
```

Indica el resultado que produce si la variable a toma un valor menor que cero. ¿Como lo arreglarías para que produjera un resultado coherente?

5. Escribe las sentencias if-else equivalentes para lo siguiente:

`h=(a>b)?(b<c)?a:b:c;`

6. Reemplaza las siguientes series de sentencias if-else con una expresión equivalente:

- (a) Usando el operador condicional ? :
 (b) Usando los operadores lógicos && y ||.

<pre>if(a>b) if(b<c) h=(c-b); else if(m>p) h= m; else h= p; else h= b;</pre>	<pre>if(a<d) h= a; else if(b<c) h= a; else if(b<d) { if(c<d) h= c; else h= d; } else h= (-1);</pre>
---	---

7. Encuentra los errores de los siguientes programas:

(a) `#include <stdio.h> /* Hay dos errores */`
`void main(void) {`
 `int i;`
 `puts("Dame el valor de i: ");`
 `scanf("%d", i);`
 `if (i=5) puts("i es igual a cinco");`
`}`

(b) `#include <stdio.h> /* Hay cinco errores */`
`void main(void) {`
 `int a=5, b=3;`
 `char ch;`
 `puts("Dame el operador (+, -, * o /);`
 `ch=getchar();`
 `switch(ch)`
 `case '+': c=a+b;`
 `case '-': c=a-b;`
 `case '*': c=a*b;`
 `case '/': c=a/b;`
 `printf(" %d %c %d = %d", a,ch,b,c);`
`}`

8. Determina la salida de los siguientes fragmentos de código:

<pre>i=6; do { i++; printf("%d ",i); } while (i<10);</pre>	<pre>i=10; while(i<10) { i--; printf("%d ",i); }</pre>
---	---

```

i=10;           i=1;
do {           while(i>0) {
    i--;       i++;
    printf("%d ",i);
    printf("%d ",i);
} while(i<10); }

```

9. ¿Cuántas veces se ejecutan cada uno de los siguientes bucles y cual es el valor final de la variable i en cada caso?

- for(i=0;i<10;i++) printf("i=%d\n",i);
- for(i=0;i<=12;i++) printf("i= %d\n",i);
- for(i=0;i<=10;i+=2) printf("i= %d\n",i);
- for(i=0;i!=10;i++) printf("i= %d\n",i);
- for(i=0.5;i>0.0;i-=0.2) printf("i= %f\n",i);
- for (i=19;i<12;i--) printf("%d ",i);

Escribe de nuevo los bucles anteriores usando while y do-while y repite el ejercicio. Intercambia el orden del printf y de la sentencia que altera el valor de la variable i en cada una de tus versiones del programa. Describe ahora que diferencias provoca esto en el programa y en el valor final de i.

10. Encuentra los errores que existen en los siguientes programas:

- /* Escribe los cuadrados de los números del 1 al 10 */
#include <stdio.h> /* Hay cuatro errores */
void main(void) {
 int i;
 for(i=1,i<10,i++)
 printf("El cuadrado de i es %d,i*i);
}
- /* Escribe valores de 1 a 100 al cubo. El paso usado es 0.3 */
#include <stdio.h> /* Hay cuatro errores */
void main(void) {
 int i;
 while(i!=100) {
 printf("%d elevado al cubo es %d /n", i*i*i);
 i += 0.3;
 }
}
- /* Determina la resistencia dado el voltaje y la corriente */
#include <stdio.h>
void main(void) {
 char input;
 puts("Cálculo de la resistencia dado el voltaje y la corriente");
 while(input=='s') {
 printf("Dame el voltaje y la corriente: ");
 scanf("%f %f", v, c);
 }
}

```

printf("La resistencia es %f, v/c);
puts("Deseas continuar (s/n)?");
input=toupper(getchar());
}
}

```

11. Escribe la salida que produce el siguiente fragmento de programa:

```

for(i=1;i<10;i++)
    for(j=1;j<10;j++)
        if(i%j==0)
            printf("%d / %d = %d\n", i, j, i/j);

```

12. Una empresa maneja códigos numéricos con las siguientes características:

- Cada código consta de cuatro dígitos.
- El primero representa a una provincia.
- El segundo el tipo de operación.
- Los dos últimos el número de la operación.

Escriba un programa que lea de teclado un número de cuatro dígitos, y posteriormente imprima en pantalla la siguiente información:

```

PROVINCIA           &
TIPO DE OPERACIÓN   &
NÚMERO DE OPERACIÓN &&

```

En caso de que el número tenga más de 4 dígitos, en lugar del mensaje anterior, habrá que imprimir en pantalla el siguiente mensaje de error:

ERROR: CÓDIGO NO VÁLIDO.

Si tiene menos de 4 dígitos se suponen 0 los primeros.

13. Con objeto de fomentar el ahorro energético, el recibo de la electricidad se elabora de forma que el precio de cada Kwh (kilowatio/hora) consumido es más caro cuanto más se consume:

- 100 Ptas de gastos fijos.
- 50 Ptas/Kwh para los primeros 100 Kwh.
- 70 Ptas/Kwh para los siguientes 150 Kwh (de 101 a 250).
- 100 Ptas/Kwh para el resto.

Elabore un programa que lea de teclado los dos últimos valores del contador (lo que marca el contador actualmente y lo que marcaba en la última lectura), y calcule e imprima en pantalla el importe total a pagar.

14. Dado un número entero que representa una determinada cantidad de dinero, escriba un programa que lea dicho número del teclado y lo desglose para saber cuantas monedas de 500, 100, 50, 25, 5 y 1 pesetas se necesitan.
15. Escriba un programa para convertir un tiempo expresado en segundos a un formato que lo represente en horas, minutos y segundos.
16. Efectuar un programa que lea los valores de 3 resistencias electrónicas (en Ohmios, Ω) conectadas en paralelo y muestre en pantalla el valor global de las 3. El valor global es calculado por la siguiente ecuación:

$$1/(1/R_1 + 1/R_2 + 1/R_3)$$

Modifique 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 global.
- (b) Que muestre también el resultado global si todas las resistencias fueran conectadas en serie (en vez de en paralelo). Recuerde que si conectamos en serie varias resistencias el resultado global es la suma de todas ellas.
17. Escriba un programa que lea un número N de teclado y a continuación imprima en pantalla:

- Su factorial.
- La suma de los N primeros términos de la serie armónica ($1 + 1/2 + \dots + 1/N$)
- Los N primeros términos de la sucesión de Fibonacci (ver ej. 19 del Tema 2).

Utilice una estructura repetitiva distinta para cada caso. ¿Cuál es más apropiada?

18. Codifique un programa que se comporte como una calculadora simple. Para ello deberá tener las siguientes características:
- Solo efectuará operaciones con dos operandos enteros.
 - Operaciones permitidas: +, -, * y /.
 - Pedirá en primer lugar el operador, y a continuación los dos operandos. Si el operador no se corresponde con alguno de los indicados se emitirá un mensaje de error.

Ejemplo:

```
Operación:      *
Operando 1:     24
Operando 2:      3
Resultado :     72
```

Modifique el programa para utilizar operandos de tipo real y para añadirle nuevas operaciones como % (operador módulo), el cálculo del tanto por ciento (usando el carácter '%'). Ejemplo: 20 % 6 calcularía el módulo de 20 entre 6, i.e. 2, y 6 P 20 calcularía el 6% de 20, que es 1.2.

19. Calculador repetitivo. Modifique el programa anterior para que se repita un número indefinido de veces. El calculador dejará de trabajar cuando se introduzca como código de operación & (ampersand). Ejemplo de ejecución:

```
Operación:      *
Operando 1:     13
Operando 2:     10
Resultado:      130
Operación:      u
***** ERROR *****
Operación:      +
Operando 1:     12
Operando 2:      3
Resultado:      15
Operación:      &
***** FIN DEL PROGRAMA *****
```

20. Realice un programa que muestre la tabla de verdad para las siguientes ecuaciones:

$$Z = \overline{A + B + AB}$$

$$Z = \overline{A + B + D + (C \cdot D) + A}$$

El operador + representa el O lógico, el operador · representa el Y lógico y la línea sobre una expresión representa la negación lógica. Resuelva el problema de forma similar al ejemplo del tema 4.

Nivel 2 : Dificultad Media

21. Escribe un programa que encuentre el mayor, el menor y la media aritmética de una lista de N números leídos por el teclado. Inicialmente el programa pedirá el número N y a continuación pedirá esos N números.
22. Efectuar un programa que lea dos números enteros y muestre en pantalla todos los números comprendidos entre dichos números, ambos incluidos. El programa no debe suponer que el primero será menor que el segundo, ni viceversa, pero sí que deberá tenerlo en cuenta para mostrar los números en orden creciente o decreciente según corresponda.
- Modifique después el programa anterior para que los números sean mostrados por filas, de forma que cada fila tenga n números. El número de números por filas, n , será leído al principio del programa. Cada número puede ir separado del siguiente por el carácter tabulador (carácter '\t').
23. Escriba un programa que resuelva completamente una ecuación de segundo grado, tanto si tiene raíces reales como imaginarias. Los coeficientes se leerán de teclado.

Ejemplo:

```
Primer coeficiente : 1
Segundo coeficiente : 0
```

Tercer coeficiente : 1
 SOLUCIONES: 0+i
 0-i

Nota: i se escribe como un caracter directamente en pantalla.

24. Confecciona un programa que lea de entrada un texto carácter a carácter hasta localizar un punto, y que al final dé como salida el número de comas encontradas, y el número de caracteres leídos.
25. Escribe un programa que determine si la cadena *abc* aparece en una sucesión de caracteres cuyo final viene dado por un punto.
26. Escribe un programa que lea un número natural *N* y dibuje un triángulo de asteriscos con base y altura *N*. Por ejemplo si *N* = 4 debería dibujarse:

```

  *
 * *
* * *
* * * *
```

27. Escribe un programa que lea un número natural *N* y un carácter. La salida debe ser un rombo compuesto del carácter y de la anchura que especifica el número *N* (diagonal horizontal). Por ejemplo, si *N* es 4 y el carácter es *, el rombo sería:

```

  *
 * *
* * *
* * * *
 * * *
  * *
   *
```

28. Escribe un programa que lea un número *N* e imprima una pirámide de números con *N* filas como en la siguiente figura para *N* = 4:

```

  1
 121
12321
1234321
```

29. Escribe un programa que lea una lista de números enteros terminada en 0, y que encuentre y escriba en la pantalla la posición de la primera y de la última ocurrencia del número 12 dentro de la lista. Si el número 12 no está en la lista, el algoritmo debería escribir 0. Por ejemplo, si el octavo número de la lista es el único 12, entonces 8 sería la primera y la última posición de las ocurrencias de 12.

30. Diseña y optimiza un programa que calcule números combinatorios.

$$C_n^m = \frac{n!}{m!(n-m)!} = \frac{n \times (n-1) \times \dots \times (n-m+1)}{m!}$$

Un número combinatorio C_n^m calcula el número de partes de *m* elementos que se pueden tomar de un conjunto de *n* elementos. Si *n* = *m*, el número es 1. Si *n* < *m* el programa deberá dar un mensaje de error ya que no se pueden tomar subconjuntos de mayor tamaño que el propio conjunto.

31. Escribe un programa para evaluar las funciones sin(*x*) y cos(*x*) dadas las siguientes series de potencias:

$$\cos(x) = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8!$$

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + x^9/9!$$

El programa deberá pedir el valor de *x* y aproximar el valor de las funciones seno y coseno calculando las anteriores series hasta el término 100. Para evitar problemas al calcular las potencias, se recomienda que se calcule cada término en función del anterior.

32. Hacer 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.
33. La Pascua de Resurrección es una fiesta de la liturgia cristiana que se celebra en Primavera (con fecha variable), en memoria de la Resurrección de Cristo. El concilio de Nicea (325) estableció a la cristiandad que la fiesta de Pascua debe celebrarse cada año el primer Domingo que sigue al decimocuarto de la lunación del equinoccio de Primavera, fijado el 21 de Marzo. Con la llegada del calendario Gregoriano, que es el que usamos en la actualidad y que fue implantado por el Papa Gregorio XIII en 1582 sustituyendo al anterior calendario, el juliano, se produjo un gran descontrol sobre cuando se debía celebrar la Pascua.

Gauss, afamado y astuto matemático, ideó un método para calcular la fecha exacta en la que celebrar la Pascua de Resurrección. Según la fórmula de Gauss la fecha de Pascua debe ser una de las dos siguientes (la única que exista de las dos):

1. El (22 + d + e) de Marzo.
2. El (d + e - 9) de Abril.

Teniendo en cuenta que:

a = año % 19
 b = año % 4
 c = año % 7
 d = (19a + M) % 30
 e = (2b + 4c + 6d + N) % 7

y donde $M=15$ y $N=6$ en el calendario juliano. En el calendario gregoriano los valores de M y N varían lentamente y, hasta el año 2100, tienen los siguientes valores: $M=24$ y $N=5$.

Efectuar, sabiendo los datos anteriores, un programa que pida un año y muestre en pantalla la fecha exacta del Domingo de Resurrección o, lo que es lo mismo, la fecha de la fiesta de la Pascua o, si lo prefieren, la fecha del último Domingo de Semana Santa. El programa deberá indicar y controlar que el año introducido sea menor o igual que el año 2100, año en el que se deben cambiar los valores de M y N .

Algunos ejemplos, son los siguientes: En 1988 y en 1994 la Pascua se celebró el 3 de Abril, en 1995 fue el 16 de Abril, en 1996 fue el 7 de Abril, en 1997 fue el 30 de Marzo y en 1998 fue el 12 de Abril.

34. Modificar el programa anterior para que muestre por pantalla también la fecha exacta del Domingo de Ramos. El Domingo de Ramos es el Domingo anterior a la Semana Santa y es justo el Domingo anterior al Domingo de Resurrección.

35. El N.I.F. (Número de Identificación Fiscal) es un número exclusivo para cada persona formado por el número del D.N.I. (Documento Nacional de Identidad) y una letra añadida. Esta letra depende directamente del número del D.N.I.

La utilidad de esta letra es poder detectar posibles errores al introducir un determinado D.N.I.. La letra es simplemente un control de errores. Por ejemplo, al D.N.I. número 12.345.678 le corresponde la letra Z. Un programa que necesite almacenar este tipo de información deberá pedir el N.I.F. (D.N.I. y su letra asociada). Si al introducir este número nos equivocamos en un dígito introduciendo 12.645.678, nos daremos cuenta de que hemos cometido algún error ya que a este último número le corresponde la letra W y no la Z.

El algoritmo para calcular esta letra, a partir del número del D.N.I. es el siguiente: Se calcula el resto de dividir el D.N.I. entre 23. Dicho resto está 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 español las letras son 28 y el siguiente número primo después del 23 es el 29. Además hay letras del español que no son ya consideradas como tales por estar formadas de dos caracteres.

Haga un programa que lea un número de D.N.I. y escriba la letra del N.I.F. que le corresponde.

Tenga en cuenta que el número del D.N.I. es un número bastante grande y no puede almacenarse en una variable de tipo `unsigned int`.

36. Modifique 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 correspondería.
37. Hacer un programa que lea una lista de cantidades de tiempo y vaya mostrando sucesivamente la suma de dichas cantidades. Cada cantidad de tiempo está formada por tres valores enteros: horas, minutos y segundos. El programa terminará cuando se introduzca un valor negativo en las horas. El programa sacará un mensaje de error si se introducen valores negativos para los minutos o los segundos.

38. Hacer un programa que calcule y muestre por pantalla una tabla con las tablas de verdad de los operadores lógicos: NOT, AND, OR, NAND, NOR y XOR. Dos columnas de esa tabla serán las variables A y B, que tomarán los valores VERDAD (TRUE) o FALSO (FALSE) y el resto de columnas serán para el resultado de cada operación lógica con esas variables.

Nivel 3 : Dificultad Avanzada

39. Calcule e imprima en pantalla los N primeros números primos, siendo N un número que se introduce por teclado. Ej: Para $N = 7$, el programa tendría que escribir los números: 1, 2, 3, 5, 7, 11, 13.
40. Escriba un programa que efectúe la conversión de un número de tipo entero positivo (`unsigned int`) en base 10 a cualquier base (menor de 10). Para ello, en primer lugar se introducirá el número a cambiar y posteriormente la base a que se desea convertir. (Sugerencia: Tenga en cuenta que los números a convertir no pueden ser mayores que el mayor número que se pueda representar con un entero positivo).

Ejemplo:

Introduzca dato: 72

Introduzca base: 4

Resultado: El número 72 en base 4 es igual a: 1020

41. Escriba un programa que calcule el valor de S para un número real X dado utilizando la siguiente serie:

$$S = 1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \frac{X^4}{4!} + \dots$$

El valor de S se calculará hasta que un término concreto $X^i/i!$ sea menor que 0.0001. Recuerde que esa fórmula sirve para calcular el valor de la exponencial de X : $S = \exp(X) = e^X$. Muestre por pantalla también el valor que devuelve la función `exp()` de la librería `math.h` pasándole X como argumento para comparar ambos valores.

42. Dada una sucesión, de longitud indeterminada, de ceros y unos, construir un programa que permita calcular el tamaño de la mayor subsecuencia ordenada de menor a mayor. La sucesión se lee desde el teclado, y el final viene dado por el número 2. Ejemplo: Para la siguiente sucesión el resultado es 4: 010010011010112.
43. Iba yo leyendo un libro por el campo cuando me percaté que, llevando leídas casi 500 páginas, se cumplía la curiosa característica de que la suma de los números de las páginas leídas es igual a la suma de los números de las páginas que me quedan por leer. De repente, tropecé con un pedrusco, el libro se me cayó y no sabía la página por la que iba leyéndolo. Haz un programa que me ayude a encontrar esta página. De camino, haz que el programa muestre también el número de páginas del libro y la suma de los números de las páginas leídas (o de las que me quedan por leer). ¡Ah! y gracias por tu ayuda.

44. El circuito eléctrico que se muestra más adelante es un filtro paso alto que atenúa las señales de baja frecuencia. La razón entre el voltaje de salida, V_{out} , con respecto al voltaje de entrada, V_{in} , puede ser expresado mediante la ecuación:

$$V_{out}/V_{in} = (X_L)/(R + X_L)$$

donde X_L representa la reactancia inductiva del inductor L y puede ser calculada con la ecuación:

$$X_L = 2\pi fL \text{ ohms}$$

y f representa la frecuencia del voltaje suministrado V_{in} . Generar una tabla de valores mostrando cómo cambia V_{out}/V_{in} variando la frecuencia suministrada y observa cómo el voltaje de salida se incrementa cuando la frecuencia f aumenta.

Si el inductor es reemplazado por un capacitor de valor C faradios, cuya reactancia capacitiva viene dada por la ecuación $X_C = 1/(2\pi fC)$, muestra que el circuito ahora se comporta como un filtro paso bajo, atenuando las señales de alta frecuencia.

Realiza un programa que pida el valor de R , L , un valor de frecuencia inicial y otro final. Se mostrará por pantalla una tabla con el valor de frecuencia y el valor V_{out}/V_{in} correspondiente desde el valor de frecuencia inicial hasta el final. De esta forma se observa cómo varía el voltaje de salida con respecto a la frecuencia. Realiza el mismo programa reemplazando el inductor por el capacitor.

