

TEMA 4: Estructuras de control

EJERCICIOS

1. La ejecución de las siguientes sentencias, donde a y b son variables enteras:

```
a = 3; b = 6;
if (!(b % 2==0) || (a<5) && (b+7<=8))
    cout << 'A';
else
    cout << 'B';
```

- a) Escribe A. b) Escribe B. c) No escribe nada. d) Origina un error durante la ejecución.

2. La ejecución de las siguientes sentencias, donde n es una variable entera:

```
do
    cin >> n;
while (n < 1 || n > 10);
if (n < 3 && n > 8)
    cout << "Dentro";
else
    cout << "Fuera";
```

- a) Escribe Dentro para valores de n en el subrango 3..8.
 b) Escribe Fuera únicamente para los siguientes valores de n : 1, 2, 9, 10.
 c) Escribe Fuera para cualquier valor de n comprendido en el subrango 1..10.
 d) Nada de lo anterior es cierto.

3. El siguiente fragmento de código emplea las variables enteras $acum$ y $cont$:

```
acum = 0;
for(cont = 1; cont <= 10; cont++)
    if (cont >= 5)
        if (acum - cont < 0)
            acum = acum + cont / 2;
        else
            acum = acum - cont / 3;
```

Indica el valor de la variable $acum$ tras la ejecución de las sentencias anteriores:

- a) 0 b) 10 c) 7 d) -3
4. Dadas las siguientes sentencias de repetición, y supuesto que s , i y n son variables de tipo entero, señalar aquellas sentencias cuyo resultado sea la *suma de los números pares comprendidos entre 1 y n* :

<pre><input type="checkbox"/> s = 0; for(i = 1; i <= n / 2; i++) s = s + i * 2; <input type="checkbox"/> s = 0; i = 2; while (i <= n) { s = s + i; i = i + 2; }</pre>	<pre><input type="checkbox"/> s = 0; i = 1; do { if (i % 2 == 0) s = s + i; i = i + 1; } while(i != n);</pre>
--	---

5. Dado el siguiente fragmento de programa, donde n, i, j son variables de tipo int:

```
j = 0; i = 5;
while (i >= 0)
{
    i = i-1;
    n = j / i;
    cout << n;
}
```

¿Qué sucede al ejecutar este código?

6. Si las variables reales m y n contienen los datos 11 y 5 respectivamente, la sentencia:

```
switch (m / n)
{
    case 1:
        cout << "uno";
        break;
    case 2:
        cout << "dos";
        break;
    default:
        cout << "ni uno ni dos";
}
```

- a) Da un error de ejecución. b) Da un error de compilación.
c) Visualiza la cadena dos. d) Nada de lo anterior es correcto.

7. Dado el siguiente bloque de código, reescríbelo primero utilizando el bucle *do...while* y después utilizando el bucle *while*, de manera que ambos funcionen exactamente igual que el original para cualquier caso.

```
cin >> n;
for(i = 1; i < n; i++)
    cout << i;
```

8. Dado el siguiente bloque de código, reescríbelo primero utilizando el bucle *while* y después utilizando el bucle *for*, de manera que ambos funcionen exactamente igual que el original para cualquier caso.

```
i = 0;
cin << n;
do
{
    i = i + 1;
    cout << i;
}
while (i <= n);
```