

Programación

Test Autoevaluación Tema 4

Autores:

M. Paz Sesmero Lorente
Paula de Toledo Heras
Fco. Javier Ordoñez Morales
Juan Gómez Romero
José A. Iglesias Martínez
José Luis Mira Peidro



Universidad
Carlos III de Madrid
www.uc3m.es



SOLUCIONES

1. Dado el siguiente programa en lenguaje C, indique qué se imprimirá en pantalla:

```
#include <stdio.h>
int main(void) {
    int i, j, r;
    for(i=0; i<2; i++) {
        for(j=0; j<3; j++) {
            r = i+j;
            if(r % 2 == 0)
                printf("%i ", r);
        }
    }
    printf("\n");
    return 0;
}
```

Solución: El comportamiento del programa se puede esquematizar en la siguiente tabla donde, para cada valor que toma la variable i se indica el valor que toman las variables j y r . En negrita se indican aquellos valores de r para los que $(r \% 2 == 0)$. Por tanto, la respuesta correcta es la opción c.

i=0	i=1
j=0	j=0
r=0	r=1
j=1	j=1
r=1	r=2
j=2	j=2
r=2	r=3

- a. No imprime nada.
- b. 0, 2, 4, 6.
- c. 0, 2, 2
- d. 0, 2, 4

2. Indique el resultado del siguiente programa en C:

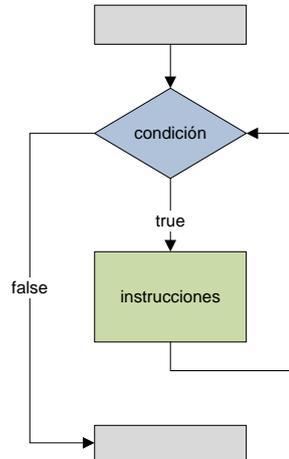
```
#include <stdio.h>
int main(void) {
    int a = 10, b;
    while(a > 0) {
        b = a - 1;
        printf("%i ", b);
        a = a - 1;
    }
    return 0;
}
```

Solución: En la primera iteración, $a=10$ y por tanto $b=9$. Así pues, el primer valor mostrado por pantalla es 9.

El bucle **while** se repite mientras $a>0$. Esto significa que el último valor que toma la variable a es 1 y que el último valor que toma b es 0. Por tanto, la opción correcta es la d.

- a. Imprime los valores: 10, 9, 8, ..., 0
- b. Imprime los valores: 10, 9, 8, ..., 1
- c. Imprime los valores: 9, 8, ..., 1
- d. Imprime los valores: 9, 8, ..., 0

3. ¿A qué instrucción de control corresponde el siguiente diagrama de flujo?



Solución: El diagrama de flujo refleja que el bloque de instrucciones se ejecuta siempre que la condición sea cierta. Por tanto, es fácil deducir que se trata de una estructura de control repetitiva. Puesto que la condición se evalúa antes de ejecutar el bloque de instrucciones se trata de una sentencia while.

- a. while @
- b. do-while
- c. if-else
- d. switch

4. ¿Cuál de las siguientes afirmaciones respecto a la instrucción de inicialización en un for es correcta?

Solución: La sintaxis general de la sentencia for es:

```

for (inicialización; expresión_lógica; actualización){
    bloque_de_instrucciones
}
  
```

y su funcionamiento el que se indica a continuación:

- Primera iteración:
 - ✓ Se ejecuta la expresión de inicialización.
 - ✓ Se evalúa la expresión_lógica y dependiendo del valor del resultado:
 - ✓ Si es cierto, se ejecuta el bloque_de_instrucciones y se vuelve al inicio del bucle
 - ✓ Si es falso, finaliza la ejecución de la sentencia.
- Segunda y sucesivas iteraciones:
 - ✓ Se ejecuta la expresión de actualización

- ✓ Se evalúa la expresión_lógica y dependiendo del resultado se ejecuta el bloque _de_instrucciones o finaliza la ejecución de la sentencia.

Por tanto, la instrucción de inicialización se ejecuta la primera vez que se llega al for tanto si se entra al bloque como si no.

- a. Se ejecuta cada vez que termina el bloque de instrucciones asociado
- b. Se ejecuta cada vez que empieza el bloque de instrucciones asociado
- c. Se ejecuta la primera vez que se llega al for, y solamente si se entra al bloque
- d. Se ejecuta la primera vez que se llega al for, tanto si se entra al bloque como si no @

5. Tras ejecutar el siguiente programa en C ¿qué valores que se muestran por pantalla?

```
#include <stdio.h>

int main(void) {
    int a=4,b=3,mayor;

    while ( (a<4) || (b<100) )
    {
        if (a>b) {
            mayor = a;
        } else {
            mayor = b;
        } else {
            mayor = 0;
        }
        a=a+mayor;
        b=b*mayor;
    }

    printf("%i %i",a,b);
}
```

Solución: Observando el código se puede comprobar que éste contiene una sentencia if - else y un else "huérfano". Por tanto, el programa tiene un error de compilación y en consecuencia, no muestra nada por pantalla.

- a. 4 3
- b. 20 144
- c. 8 12
- d. Ninguna de las respuestas anteriores es correcta @

6. Indique cuál de las siguientes afirmaciones es verdadera:

- a. **La estructura de control "while" se utiliza cuando no se sabe el número de repeticiones de código que va a producirse.** Cierto.
- b. **La estructura de control "while" necesita un contador que indica cuándo debe abandonarse el bucle.** Falso. En una sentencia while el bloque de instrucciones se ejecuta mientras la expresión lógica sea cierta.
- c. **Si escribimos while (0){} provocaremos la ejecución de un bucle infinito.** Falso. La condición (0) siempre es falsa (vale 0) por tanto nunca se ejecutará el bloque de instrucciones.
- d. **Si escribimos while (1){} la ejecución del programa se detendrá.** Falso. La condición (1) siempre es cierta (toma un valor distinto de 0), por tanto while(1){} provoca un bucle infinito.

7. ¿Qué imprime el siguiente programa en pantalla?

```
#include <stdio.h>
int main(void) {
    int i=0, r;
    while(i<3) {
        switch(i) {
            case 0:
                r = i;
                break;
            case 1:
                r = i+1;
                break;
            case 2:
                r = i+2;
            case 3:
                r = i+3;
        }
        i++;
    }
    printf("%i ", r);
    return 0;
}
```

- a. 6
- b. 5@
- c. 4
- d. 3

Solución: El comportamiento del programa se puede esquematizar en la siguiente tabla donde, para cada valor que toma la variable *i* se indica el o los valores que toma la variable *r*. En negrita se indica el valor que toma *r* tras finalizar el bucle while. Por tanto, la respuesta correcta es la opción b.

i=0	i=1	i=2
r=1	r=2	r=4*
		r=5

(*) Se ejecutan todas las sentencias hasta encontrar un *break* o hasta llegar al final de la sentencia *switch*.

8. Marque una de las siguientes afirmaciones como verdadera:
- En la estructura de control "switch", el código incluido en cada "case" debe necesariamente terminar con "break". Falso, si se omite la sentencia *break* el compilador entiende que también debe ejecutarse el bloque del case siguiente.
 - El bloque "default" se ejecuta si el valor de la variable selectora no coincide con el valor de algún bloque. Cierto
 - Siempre es más conveniente usar la estructura de control "switch" que la estructura de control "if". Falso. La idoneidad de una u otra estructura de control alternativa depende del problema a resolver.
 - La estructura de control "switch" puede utilizarse para sustituir una estructura de control "for". Falso. la sentencia *switch* es una sentencia de control alternativa mientras que *for* es una sentencia de control repetitiva. Por tanto, el funcionamiento de ambas sentencias es muy distinto y no son intercambiables.
9. Indique cuál de las siguientes afirmaciones respecto a la estructura de control *do-while* es falsa:...
- La expresión lógica se evalúa después de ejecutar el bloque de instrucciones. Cierto
 - La condición de control debe ir encerrada entre paréntesis. Cierto
 - El bloque de instrucciones se ejecuta como mínimo una vez. Cierto
 - do* y *while* delimitan el bloque de instrucciones y, por tanto, no es necesario el uso de llaves. Falso. Si el bloque de instrucciones está formado por más de una sentencia, todas ellas deben ir encerradas entre llaves. En caso contrario el compilador detecta un error de sintaxis..
10. Indique cuál de las siguientes afirmaciones sobre estructuras de control anidadas es cierta:
- En el anidamiento de instrucciones de control repetitivas es recomendable que un bucle interno modifique la expresión de control del bucle externo. Falso.
 - Las sentencias *if* anidadas implementan decisiones que implican una o dos alternativas. Falso. Las decisiones que implican una o dos alternativas se implementan con las sentencias *if* o *if-else* respectivamente.
 - Un bucle interno que se controle mediante el decremento de una variable debe repetirse sólo una vez en cada iteración del bucle externo. Falso
 - Ninguna de las respuestas anteriores es cierta. Cierto