

# PROGRAMACIÓN I

**Comenzado el** sábado, 17 de noviembre de 2018, 23:02

**Estado** Finalizado

**Finalizado en** sábado, 17 de noviembre de 2018, 23:55

**Tiempo empleado** 52 minutos 55 segundos

**Puntos** 45,50/50,00

**Calificación** 9,10 de 10,00 (91%)

## Pregunta 1

Correcta

Puntúa 1,00  
sobre 1,00

Si en una función main aparecen las siguientes instrucciones

```
int *p;
```

```
int i;
```

```
int k;
```

```
i = 42;
```

```
k = i;
```

```
p = &i;
```

¿Qué instrucción cambiará el valor de **i** a 75?

Seleccione una:

- ☒ \*p = 75; ✓
- ☐ p = 75;
- ☐ k = 75;

EXPLICACIÓN:

p apunta a la dirección de i, por lo tanto \*p equivale a i.

La respuesta correcta es: \*p = 75;

## Pregunta 2

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué mostrará por pantalla el siguiente programa?

```
#include <stdio.h>
```

```
int main() {  
    int i;
```

```
    for (i = 70; i; i -= 2) {  
        printf("%d\n", i);  
        i -= 2;  
    }
```

```
    return 0;  
}
```

Seleccione una:

- ☐ Se imprimen todos los números del 70 al 2 de cuatro en cuatro.
- ☐ Se imprimen todos los números del 70 al 2 de dos en dos.
- ☒ Se entra en un bucle infinito. ✓

Respuesta correcta

SOLUCIÓN:

tras cada iteración del bucle for i decrementa su valor 4 unidades:

iteración 1: i = 66

iteración 2: i = 62

...

iteración 17: 2

como 70 no es divisible por 4 i nunca llegará a valer 0 exactamente, por lo que la condición del bucle no se cumplirá y se creará un bucle infinito.

La respuesta correcta es: Se entra en un bucle infinito.

### Pregunta 3

Correcta

Puntúa 1,00  
sobre 1,00

Dado el siguiente programa, ¿qué se mostraría por pantalla?

```
#include <stdio.h>
```

```
#define X 100  
#define Y 50  
#define Z 10  
#define TOTAL X + (Y + Z)
```

```
int main() {  
    printf("%d\n", TOTAL*2);  
    return 0;  
}
```

Seleccione una:

- ☐ 170
- ☐ 320
- ☒ 220 ✓

Respuesta correcta

SOLUCIÓN:

El compilador sustituye las macros literalmente, por lo que la operación que se imprime es  $100 + (50 + 10) * 2$ , cuyo resultado, aplicando la prioridad de operadores, es 220.

La respuesta correcta es: 220

#### Pregunta 4

Correcta

Puntúa 1,00  
sobre 1,00

Si en una función main aparecen las siguientes instrucciones

```
int *pnumero;
```

```
int num1, num2;
```

```
char *pchar;
```

```
char letra1;
```

```
num1 = 2;
```

```
num2 = 5;
```

```
letra1 = 'a';
```

```
pchar = &letra1;
```

```
*pchar = 'B';
```

¿Cómo obtengo el valor de **letra1**?

Seleccione una:

- ☐ Con &letra1 o pchar
- ☒ Con letra1 o \*pchar ✓
- ☐ Con \*letra1 o &pchar

EXPLICACIÓN:

Como pchar apunta letra1, un equivalente a letra1 es \*pchar

La respuesta correcta es: Con letra1 o \*pchar

### Pregunta 5

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>
int main ()
{
    int i;

    for (i=1; i<4; i++) {
        if (i%2)
            printf ("%d", i);
    }
    return 0;
}
```

Seleccione una:

- ☐ 24
- ☐ 1
- ☒ 13 

#### SOLUCIÓN:

Recuerda que `if (i%2)` es equivalente a `if (i%2 != 0)`.

- La variable `i` se inicializa a 1.
- Como `1<4` es cierto, se ejecuta el cuerpo del bucle:
  - Como `1%2` es 1, es decir, cierto, se escribe `i`, es decir, 1.
  - Se incrementa `i`, que pasa a valer 2.
- Como `2<4` es cierto, se ejecuta el cuerpo del bucle:
  - Como `2%2` es 0, es decir, falso, no se escribe el valor de `i`.
  - Se incrementa `i`, que pasa a valer 3.
- Como `3<4` es cierto, se ejecuta el cuerpo del bucle:
  - Como `3%2` es 1, es decir, cierto, se escribe `i`, es decir 3.
  - Se incrementa `i`, que pasa a valer 4.
- Como `4<4` es falso, se termina el bucle.

Por lo tanto, el programa escribe 1 3.

La respuesta correcta es: 13

### Pregunta 6

Correcta

Puntúa 1,00  
sobre 1,00

Estudia las siguientes declaraciones:

```
typedef struct {  
    int c;  
} H;
```

```
typedef struct {  
    int a;  
} N;
```

```
typedef struct {  
    int y;  
    N b;  
    H s;  
} V;
```

```
typedef struct {  
    int z;  
    V c;  
} M;
```

```
typedef struct {  
    V x;  
    M a;  
    V r;  
} U;
```

U mi;

¿Cuál de los siguientes accesos es correcto?

Seleccione una:

- ☐ mi.a.c.x
- ☐ mi.r.y.c
- ☒ mi.x.b.a ✓

mi es de tipo U

En mi, dentro de la estructura U, se puede referenciar al miembro x, que es de tipo V

En x, dentro de la estructura V, se puede referenciar al miembro b, que es de tipo N

En b, dentro de la estructura N, se puede referenciar al miembro a.

La respuesta correcta es: mi.x.b.a

**Pregunta 7**

Correcta

Puntúa 1,00  
sobre 1,00

Dada la siguiente estructura de datos:

```
#define DIM 30
```

```
typedef struct {  
    int dia; /* Dia de nacimiento */
```

```
  
    int mes; /* Mes de nacimiento */  
    int anyo; /* Anyo de nacimiento */  
} Fecha;
```

```
typedef struct {  
    char nombre [DIM];
```

```
  
    char contra [DIM];
```

```
  
    Fecha fecha;  
} Usuario;
```

```
int main(){
```

```
  
    Usuario usuario;  
    /* Solicitar día de nacimiento */  
    return 0;  
}
```

Se desea solicitar para un usuario el día de nacimiento.Cuál de las siguientes instrucciones son correctas:

Seleccione una:

- ☐ printf ("Introduzca día de nacimiento: ");  
scanf ("%d", &usuario.dia);
- ☐ printf ("Introduzca día de nacimiento: ");  
scanf ("%d", &fecha.dia);
- ☒ printf ("Introduzca día de nacimiento: ");  
scanf ("%d", &usuario.fecha.dia); ✓

**SOLUCIÓN:**

Se ha de acceder a la variable usuario, a continuación a su miembro fecha y dentro de este al miembro dia.

La respuesta correcta es: printf ("Introduzca día de nacimiento: ");  
scanf ("%d", &usuario.fecha.dia);

**Pregunta 8**

Correcta

Puntúa 1,00  
sobre 1,00

Dada las siguiente estructuras de datos y declaraciones:

```
#define DIM 30
```

```
#define U 5
```

```
typedef struct  
{  
int dia; /* Dia de nacimiento */  
int mes; /* Mes de nacimiento */  
int anyo; /* Anyo de nacimiento */  
} Fecha;
```

```
typedef struct  
{  
char nombre [DIM];  
char contra [DIM];  
Fecha fecha;  
} Usuario;
```

```
typedef struct  
{  
int numUsuarios;  
Usuario usuario [U];  
} Sesion;
```

```
Sesion sesion;
```

Se desea introducir el año de nacimiento del cuarto usuario. Seleccione la opción correcta:

Seleccione una:

- ☐ scanf ("%d", &sesion.usuario.fecha.anyo[4]);
- ☐ scanf ("%d", sesion.usuario[4].fecha.anyo);
- ☒ scanf ("%d", &sesion.usuario[3].fecha.anyo); ✓

**SOLUCIÓN:**

Dentro de la variable sesion hay que acceder a su tabla usuario. En la tabla el cuarto usuario se guarda en la posición 3, y dentro de esta posición se accede a fecha y anyo.

La respuesta correcta es: scanf ("%d", &sesion.usuario[3].fecha.anyo);



**Pregunta 9**

Incorrecta

Puntúa -0,50  
sobre 1,00

¿Qué formato se utiliza para mostrar por pantalla el contenido de una variable de tipo double?

Seleccione una:

- ☐ %d
- ☒ %f **✗**
- ☐ %lf

SOLUCIÓN:

%f se utiliza para variables de tipo float y %d para variables de tipo int

La respuesta correcta es: %lf

**Pregunta 10**

Incorrecta

Puntúa -0,50  
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>
```

```
int main ()  
{
```

```
    int n=4;
```

```
    while (n )  
        n = n%3;
```

```
    printf ( "%d", n);
```

```
    return 0;  
}
```

Seleccione una:

- ☐ 0
- ☐ Ninguna de las otras
- ☒ 1 **✗**

SOLUCIÓN:

El programa no imprimiría nada ya que se queda en un bucle infinito pues n tomaría siempre el valor 1 nunca tendría el valor de salida del bucle.

La respuesta correcta es: Ninguna de las otras

### Pregunta 11

Correcta

Puntúa 1,00  
sobre 1,00

Analiza el siguiente programa:

```
#include <stdio.h>
```

```
#define SIZE 100
```

```
int main(){
```

```
    char nombre[SIZE];
```

```
    printf("Introduce tu nombre: ");
```

```
    scanf("%s", nombre);
```

```
    printf("%s", nombre);
```

```
    return 0;
```

```
}
```

Si durante la ejecución, el usuario escribe el nombre "*Bruno Perez*" cuando se lo indican por pantalla, ¿qué salida emitirá el segundo *printf*?

Seleccione una:

- ☐ Bruno Perez
- ☐ Dará un error en tiempo de ejecución
- ☒ Bruno ✓

Respuesta correcta

La función *scanf* no recoge los espacios, por lo que acabará de leer la cadena en el espacio entre *Bruno* y *Perez*, almacenando únicamente el valor "Bruno" en la variable.

La respuesta correcta es: Bruno

### Pregunta 12

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué efecto tiene la instrucción *break* cuando aparece dentro de un bucle?

Seleccione una:

- ☒ Abandona el bucle y sigue ejecutando la siguiente instrucción tras el bucle ✓
- ☐ Termina la ejecución del programa
- ☐ Permite volver a evaluar la condición de salida del bucle

SOLUCIÓN:

La instrucción *break* en un bucle hace que se abandone el bucle y se siga ejecutando la siguiente instrucción tras el bucle

La respuesta correcta es: Abandona el bucle y sigue ejecutando la siguiente instrucción tras el bucle

### Pregunta 13

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué muestra el siguiente programa si el usuario introduce un "2"?

```
#include <stdio.h>

typedef enum {altaUsuario = 1, nuevaSesion, salir} MenuInicial;

int main()
{
    int opcion;
    printf("Seleccione una opcion:\n");
    printf("1.- Dar de alta un nuevo usuario\n");
    printf("2.- Abrir nueva sesion\n");
    printf("3.- Salir\n");

    scanf("%d", &opcion);

    switch(opcion)
    {
        case altaUsuario:
            printf("Ha seleccionado dar de alta un nuevo usuario\n");
            break;
        case nuevaSesion:
            printf("Ha seleccionado abrir una nueva sesion\n");
        case salir:
            printf("Ha seleccionado salir\n");
            break;
        default:
            printf("Ha seleccionado una opcion incorrecta\n");
            break;
    }
    return 0;
}
```

Seleccione una:

- ☒ Ha seleccionado abrir una nueva sesion  
Ha seleccionado salir ✓
- ☐ Nada, ya que no compila
- ☐ Ha seleccionado abrir una nueva sesión

#### SOLUCION:

Al haber pulsado la tecla 2, se entrará en el "case nuevaSesion" y se mostrará el mensaje correspondiente. Dado que no hay ningún break, se mostrarán todos los mensajes hasta encontrarlo, por lo tanto, también se mostrará por pantalla el mensaje "Ha seleccionado salir"

La respuesta correcta es:

Ha seleccionado abrir una nueva sesion  
Ha seleccionado salir

**Pregunta 14**

Correcta

Puntúa 1,00  
sobre 1,00

La definición (declaración) de variables en un programa "C":

Seleccione una:

- ☐ No es obligatoria, se pueden utilizar las variables sin declararlas
- ☐ Las variables se pueden declarar o no, dependiendo del programador
- ☒ Si no declaramos una variable antes de usarla, el programa no funcionará ✓

SOLUCIÓN:

Las variables sólo se pueden utilizar una vez declaradas. La declaración de variables es obligatoria para poder usarse.

La respuesta correcta es: Si no declaramos una variable antes de usarla, el programa no funcionará

**Pregunta 15**

Correcta

Puntúa 1,00  
sobre 1,00

Dado el siguiente fragmento de código

```
int i;
```

```
i++;  
printf("%d\n",i);
```

¿Cuál es el valor que se escribe por pantalla?

Seleccione una:

- ☐ 0
- ☐ 1
- ☒ Un valor indefinido ✓

SOLUCIÓN:

A la variable i no se le asigna ningún valor, por lo que no se puede determinar qué valor contiene.

La respuesta correcta es: Un valor indefinido

### Pregunta 16

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>
int main ()
{
    int i, cont;
    int datos[]={1,2,0};

    for (i=0, cont=0; i<3; i++)
        if (i%2)
            cont+=datos[i];

    printf ("%d", cont);

    return 0;
}
```

Seleccione una:

- ☒ 2 ✓
- ☐ 1
- ☐ 3

#### SOLUCIÓN:

- La variable i se inicializa a 0.
- La variable cont se inicializa a 0.
- Como  $i < 3$  es cierto, se ejecuta el cuerpo del bucle.
  - Como  $i \% 2 = 0 \% 2 = 0$  es falso, no se incrementa cont.
- Se incrementa i, que pasa a valer 1.
- Como  $i < 3$  ( $1 < 3$ ) es cierto, se ejecuta el cuerpo del bucle.
  - Como  $i \% 2 = 1 \% 2 = 1$  es cierto, se incrementa cont con  $\text{datos}[1]=2$ , es decir, cont pasa a valer 2.
- Se incrementa i, que pasa a valer 2.
- Como  $i < 3$  ( $2 < 3$ ) es cierto, se ejecuta el cuerpo del bucle.
  - Como  $i \% 2 = 2 \% 2 = 0$  es falso, no se incrementa cont.
- Se incrementa i, que pasa a valer 3.
- Como  $i < 3$  ( $3 < 3$ ) es falso, se acaba el bucle.
- Se escribe cont, es decir, 2.

La respuesta correcta es: 2

### Pregunta 17

Correcta

Puntúa 1,00  
sobre 1,00

El formato “%4d”:

Seleccione una:

- ☐ Escribe un entero a la izquierda de un campo de anchura 4
- ☐ Escribe un entero utilizando 4 dígitos
- ☒ Escribe un entero a la derecha de un campo de anchura 4 ✓

SOLUCIÓN:

%4d escribe un entero a la derecha de un campo de anchura 4, tal y como se especifica y solicita en el enunciado del ejercicio 3 de la práctica 2.

La respuesta correcta es: Escribe un entero a la derecha de un campo de anchura 4

### Pregunta 18

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuál es la salida de la instrucción `printf("*****\n** hola **\n*****\n");`?

Seleccione una:

- ☐ \*\*\*\*\*
- ☐ \*\*\*\*\* hola \*\*\*\*\*
- ☒ \*\*\*\*\*  
\*\* hola \*\*  
\*\*\*\*\* ✓

SOLUCIÓN:

Los saltos de línea hacen que se muestre un mensaje con tres líneas

La respuesta correcta es:

```
*****
** hola **
*****
```

**Pregunta 19**

Correcta

Puntúa 1,00  
sobre 1,00

Dado el siguiente programa:

```
#include <stdio.h>

#define N 4

int main()
{
    int contador=1;
    int x, maximo=-10000, minimo=10000;
    while (contador<N)
    {

        printf("x= ");

        scanf("%d",&x);

        if (x>maximo)
            maximo=x;

        if (x<minimo)
            minimo=x;

        contador++;
    }

    printf("\nEl maximo de los N= %d números es %d\n", N, maximo);
    printf("\nEl minimo de los N= %d números es %d\n", N, minimo);

    return 0;
}
```

Si introducimos los números: 5, 8 y 12:

Seleccione una:

- ☐ El programa muestra como mínimo el 5 y como máximo el 14.
- ☐ El programa no muestra ningún resultado.
- ☒ El programa muestra como mínimo el 5 y como máximo el 12. ✓

**SOLUCIÓN:**

En la variable "maximo" se va guardando el mayor valor en cada iteración. En la variable "minimo" se va almacenando el minimo valor en cada iteración. El contador no incrementa el valor de la variable x.

La respuesta correcta es: El programa muestra como mínimo el 5 y como máximo el 12.

**Pregunta 20**

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué comandos habría que añadir para que el siguiente programa calculase si un número es primo o no y lo mostrase por pantalla?

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int num;
    int divisible = 0;
    int i;
    printf("Introduzca un numero: ");
    scanf("%d", &num);

    // Comandos para calcular si es primo
```

```
    if (divisible)
        printf("No es primo\n");
    else
        printf("Es primo\n");

    return 0;
}
```

Selecione una:



```
for (i = 2; i < num; i++) {
    if (num % i==0) {
        divisible = 1;
        break;
    }
}
```



```
for (i = 0; i < num; i++) {
    if (num % i==0) {
        divisible = 1;
    }
}
```





```
for (i = 2; i < num; i++) {  
    if (num % i==0) {  
        divisible = 0;  
        break;  
    }else {  
        divisible = 1;  
        break;  
    }  
}
```

#### SOLUCIÓN:

Para que un número sea primo, sólo debe ser divisible por 1 y por sí mismo. Por lo tanto, si un número es divisible por otro (diferente a él mismo y 1), entonces, no será primo. De esta forma, el código para saber si un número es primo o no debe recorrer todos los números menores a él, comenzando desde 2 (se da por hecho que es divisible por 1). Si en algún momento del bucle el número que saber si es primo es divisible por el número que se está probando, podemos terminar el bucle y afirmar que el número introducido no es primo.

La respuesta correcta es:

```
for (i = 2; i < num; i++) {  
    if (num % i==0) {  
        divisible = 1;  
        break;  
    }  
}
```

#### Pregunta 21

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuál es la diferencia entre la sentencia while() y la sentencia do while()?

Seleccione una:

- ☒ En la sentencia do while() siempre se ejecuta al menos una vez el cuerpo del bucle. ✓
- ☐ La sentencia do while () es mas legible, los programas quedan mas sencillos de entender.
- ☐ La sentencia do while() es más segura, el programa es más difícil que se quede en un bucle infinito.

La respuesta correcta es: En la sentencia do while() siempre se ejecuta al menos una vez el cuerpo del bucle.

## Pregunta 22

Correcta

Puntúa 1,00  
sobre 1,00

4.¿Qué le falta al siguiente código para que funcione correctamente?

```
#include <stdio.h>
```

```
int main(){  
int edad, inicial, actual;  
printf("Introduzca su anyo de nacimiento: ");  
scanf("%d", &inicial);  
edad=actual-inicial;  
printf("Su edad es: %d", edad);  
return 0;  
}
```

Seleccione una:

- ☐ Nada, el programa funciona correctamente
- ☒ Inicializar la variable "actual" a un valor ✓
- ☐ Un "\n" después de los printf

SOLUCIÓN:

En la instrucción `edad=actual-inicial;` se está utilizando la variable `actual` cuando no se le ha asignado ningún valor previamente.

La respuesta correcta es: Inicializar la variable "actual" a un valor

### Pregunta 23

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>
```

```
void f (int * p)
{
    *p=1;
    printf ( "%d", *p);
}
```

```
int main()
{
    int x=2;
```

```
f (&x);
printf( "%d", x);
```

```
return 0;
}
```

Seleccione una:

- ☐ Ninguna de las otras
- ☒ 11 ✓
- ☐ 12

#### EXPLICACIÓN:

El primer printf que se ejecuta es el de la función f y escribe \*p, que vale 1. El segundo printf que se ejecuta es el de main, que escribe el valor de la x de main.

La llamada f(&x) hace que p valga &x. Por lo tanto, \*p=1 es lo mismo que x=1. Este es el valor que se escribe desde main.

La respuesta correcta es: 11

## Pregunta 24

Correcta

Puntúa 1,00  
sobre 1,00

Dado el siguiente programa:

```
#include <stdio.h>
```

```
int menuInicial() {  
    int opcion;  
    do {  
        printf("\n");  
        printf("1. Dar de alta a un nuevo usuario.\n");  
        printf("2. Iniciar una sesion nueva.\n");  
        printf("3. Salir del programa.\n\n");  
        printf("Elija una opcion: ");  
        scanf("%d", &opcion);  
        if (opcion < 1 || opcion > 3)  
            printf("Opcion incorrecta.\n");  
    } while (opcion < 1 || opcion > 3);  
    return opcion;  
}  
  
int main() {  
    opcion = menuInicial();  
    printf("La opcion elegida es la %d", opcion);  
    return 0;  
}
```

Si introduzco por teclado el valor 1, ¿Qué muestra el programa por pantalla?

Seleccione una:

- ☐ La opcion elegida es la 1.
- ☐ Nada.
- ☒ El programa no compila porque tiene un error. ✓

### EXPLICACIÓN:

La variable opcion sólo se ha declarado dentro de la función menuInicial, por lo que no se puede utilizar dentro de la función main si no se declara también en esta última.

La respuesta correcta es: El programa no compila porque tiene un error.

### Pregunta 25

Correcta


Puntúa 1,00  
sobre 1,00

Después de ejecutar la siguiente instrucción

```
int datos [5] = {20,39,48,22,1};
```

¿qué valor tendrá la componente datos[3-1] de la tabla?

Seleccione una:

- ☐ 39
- ☐ 22
- ☒ 48 

SOLUCIÓN:

datos[3-1] es igual a datos[2], que corresponde con el valor de la tercera posición de la tabla:  
48

La respuesta correcta es: 48

**Pregunta 26**

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuántas veces se mostrará el menú de inicio antes de terminar el programa si el usuario va a teclear como opción 1?

```
#include <stdio.h>
```

```
typedef enum {altaUsuario=1, nuevaSesion, salir} MenuInicial;
```

```
int main() {
```

```
    int opcion=salir;
```

```
while (opcion!=salir)
```

```
{
```

```
printf ("\n");
```

```
printf ("1. Dar de alta a una nueva persona.\n");
```

```
printf ("2. Iniciar una sesion para un usuario.\n");
```

```
printf ("3. Salir del programa.\n\n");
```

```
printf ("Elija una opcion: ");
```

```
scanf ("%d", &opcion);
```

```
switch (opcion)
```

```
{
```

```
case altaUsuario:
```

```
printf("\nAlta usuario\n\n");
```

```
break;
```

```
case nuevaSesion:
```

```
printf("\nNueva sesion.\n\n");
```

```
break;
```

```
case salir:
```

```
printf ("\nAdios.\n\n");
```

```
break;
```

```
default:
```

```
printf ("\nOpcion incorrecta.\n\n");
```



```
}
```

```
}
```

```
return 0;
```

```
}
```

Seleccione una:

- ☐ Una.
- ☐ Hasta que el usuario introduzca '3'.
- ☒ Ninguna. ✓

SOLUCIÓN:

La variable `opcion` se inicializa al valor `salir`, y la condición de mantenimiento del bucle es que `opcion != salir`. No se llega a ejecutar una sola vez ya que en la primera iteración no se cumple la condición.

La respuesta correcta es: Ninguna.

### Pregunta 27

Incorrecta

Puntuación -0,50  
sobre 1,00

Si queremos obtener la siguiente salida:

`%% HOLA %%`

¿Qué comando debemos utilizar?

Seleccione una:

- ☒ `printf("\% \% HOLA \% \");` ✗
- ☐ `printf("%% HOLA %%");`
- ☐ `printf("%%%% HOLA %%%");`

SOLUCIÓN:

Para mostrar un % por pantalla dentro del printf se han de escribir dos % consecutivos

La respuesta correcta es: `printf("%%%% HOLA %%%");`

## Pregunta 28

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué muestra por pantalla el siguiente programa?

```
#include <stdio.h>
void f2(int c)
{
    c=4;
}
void f1(int *b)
{
    *b=2;
    f2(*b);
}
int main()
{
    int a=9;
    f1(&a);
    printf("`%d`", a);
    return 0;
}
```

Seleccione una:

- ☐ 9
- ☐ 4
- ☒ 2 

### EXPLICACIÓN:

La primera función recibe la variable por referencia mientras que la segunda la recibe por valor, por lo que sólo se mantiene la modificación realizada en la primera función.

La respuesta correcta es: 2

### Pregunta 29

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué se imprime por pantalla al ejecutar el siguiente código?

```
char func1(char c) {
```

```
    return 'c';
```

```
}
```

```
int main() {
```

```
    printf("%c", func1('f'));  
    return 0;
```

```
}
```

Seleccione una:

- ☒ c ✓
- ☐ cf
- ☐ f

#### EXPLICACIÓN:

Se imprime lo que devuelve la función func1, esto es, el carácter c.

La respuesta correcta es: c

**Pregunta 30**

Correcta

Puntúa 1,00  
sobre 1,00

1.¿Cómo completarías la instrucción scanf en el siguiente programa?

```
#include <stdio.h>
```

```
int main() {
```

```
    int numero;
```

```
    printf( "Introduce un numero: ");
```

```
    scanf( "%d",
```

```
    printf( "\nHas introducido el numero %d\n", numero);
```

```
    return 0;
```

```
}
```

Seleccione una:

- ☒ &numero); ✓
- ☐ );
- ☐ numero);

**SOLUCIÓN:**

Se ha de especificar la variable donde se va a guardar la información precedida por &  
La respuesta correcta es: &numero);

**Pregunta 31**

Correcta

Puntúa 1,00  
sobre 1,00

Para recoger un número entero introducido por el usuario...

Seleccione una:

- ☐ Se debe utilizar scanf("Introduzca un numero: %d", variable)
- ☒ Se debe utilizar scanf("%d", &variable); ✓
- ☐ Se debe utilizar scanf("%d", variable);

**SOLUCIÓN:**

Sólo la instrucción scanf("%d", &variable); lee un valor entero correctamente incluyendo & delante del nombre de la variable.

La respuesta correcta es: Se debe utilizar scanf("%d", &variable);

### Pregunta 32

Correcta

Puntúa 1,00  
sobre 1,00


¿Qué escribe el siguiente programa?

```
#include <stdio.h>
```

```
int f (char a, char c) {  
    if (a == c)  
        return 1;  
    else  
        return 0;  
}  
int main () {  
    char car='i';  
    if (f('b','c'))  
        printf ("0");  
    else  
        printf ("1");
```

```
    return 0;  
}
```

Seleccione una:

- ☒ 1 
- ☐ Ninguna de las otras
- ☐ 0

#### EXPLICACIÓN:

La instrucción:

```
if (f('b','c')) ...
```

es igual que

```
if (f('b','c') != 0) ...
```

En la llamada f('b', 'c'), como los caracteres 'b' y 'c' son distintos, la función f devuelve 0.

Como 0 es falso, se ejecuta el caso else de la instrucción if-else, es decir, se escribe 1.

La respuesta correcta es: 1

### Pregunta 33

Correcta

Puntúa 1,00  
sobre 1,00

Dada la siguiente definición

```
char s[7]="Hola";
```

¿Cuántos bytes ocupa la variables s?

Seleccione una:

- ☐ 5
- ☐ 4
- ☒ 7 ✓

SOLUCIÓN:

La variable s ocupa 7 bytes, que son los definidos en la declaración, aunque sólo se ocupen con datos los 5 primeros.

La respuesta correcta es: 7

### Pregunta 34

Correcta

Puntúa 1,00  
sobre 1,00

En un programa en C se utiliza el siguiente tipo para almacenar fechas:

```
typedef struct {  
    int dia;  
    int mes;  
    int anno;  
} Fecha;
```

A lo largo de un programa que utiliza dos variables de tipo *Fecha*, *fecha1* y *fecha2*, aparece la siguiente sentencia *if* seguida de una llamada a la función *printf()*:

```
if ((fecha1.anno >= 0) && (fecha2.anno >= 0) && !(fecha1.anno - fecha2.anno))  
    printf("%s\n", mensaje);
```

¿Qué debería contener el string mensaje para describir las fechas comparadas?

Seleccione una:

- ☐ "Los años no son correctos"
- ☐ "Los años son distintos"
- ☒ "Los años son iguales" ✓

Respuesta correcta

El operador lógico ! devuelve TRUE si la expresión aritmética a la que se aplica vale 0 y devuelve FALSE en otro caso.

La respuesta correcta es: "Los años son iguales"

### Pregunta 35

Correcta

Puntúa 1,00  
sobre 1,00

Dado el siguiente tipo abstracto de dato Usuario

```
typedef struct {  
    char nombre [20];  
    char contra [20];  
    Fecha fecha;  
} Usuario;
```

Se desea declarar una variable de tipo "Usuario".Cuál de las siguientes declaraciones es correcta:

Seleccione una:

- ☐ usuario usuario;
- ☐ Usuario nombre user;
- ☒ Usuario users; ✓

#### SOLUCIÓN:

Las respuestas erróneas utilizan usuario en minúscula para el tipo de dato y dan dos nombres (nombre y user) a dos variables que no separan por comas.

La respuesta correcta es: Usuario users;

**Pregunta 36**

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuál es el valor final de los semáforos del siguiente código?



```

#include <stdio.h>
typedef enum {rojo = -1, verde, amarillo} Color;

typedef struct {
    int numCoches; // Numero de coches esperando a que se abra el semaforo
    int estado;    // Estado del semaforo: rojo, verde o amarillo
} Semaforo;

int main()
{
    Semaforo sem1, sem2;
    sem1.estado = verde;
    sem1.numCoches = 0;
    sem2.estado = rojo;
    sem2.numCoches = 1;
    switch (sem1.estado)
    {
        case rojo:
            sem2.estado = verde;
            sem2.numCoches = 0;
            sem1.numCoches++;

        case verde:
            sem2.estado = rojo;
            sem1.estado=amarillo;
            sem1.numCoches = 0;
            sem2.numCoches++;

        case amarillo:
            if (sem1.numCoches >= sem2.numCoches)
            {
                sem2.estado = rojo;
                sem1.numCoches = 0;
                sem2.numCoches++;
            }
            else
            {
                sem2.estado = verde;
                sem2.numCoches = 0;
                sem1.numCoches++;
                sem1.estado=rojo;
            }
            break;
        default:
            printf("ERROR. Estado desconocido. Reiniciando semaforos\n");
            sem1.estado = verde;
            sem1.numCoches = 0;
            sem2.estado = rojo;
            sem2.numCoches = 1;
            break;
    }

    return 0;
}

```

```
}
```

Seleccione una:

☐

El semáforo 1 está en verde y no tiene coches esperando.  
El semáforo 2 está en rojo y tiene 2 coches esperando

☒

El semáforo 1 está en rojo y tiene un coche esperando.  
El semáforo 2 está en verde y no tiene coches esperando ✓

☐

El semáforo 1 está en amarillo y no tiene coches esperando.  
El semáforo 2 está en rojo y tiene 2 coches esperando

#### SOLUCIÓN:

Dado el estado inicial de los semáforos, se entraría al "case verde", actualizando los valores a:

```
sem1.estado=amarillo  
sem1.numCoches=0  
sem2.estado=rojo  
sem2.numCoches=2
```

Dado que no hay un break, se entraría también al "case amarillo" y, dado que el numero de coches en sem2 es mayor que en sem1, se alterarían los valores a:

```
sem1.estado=rojo  
sem1.numCoches=1  
sem2.estado=verde  
sem2.numCoches=0
```

Siendo estos los valores finales

La respuesta correcta es:

El semáforo 1 está en rojo y tiene un coche esperando.

El semáforo 2 está en verde y no tiene coches esperando

### Pregunta 37

Correcta

Puntúa 1,00  
sobre 1,00

Si queremos que se modifique el valor de la variable que pasamos como argumento a una función, ¿qué debemos hacer?

Seleccione una:

- ☐ Pasar la variable por valor.
- ☒ Pasar la variable por referencia. ✓
- ☐ Devolver la variable como retorno.

#### EXPLICACIÓN:

El paso de una variable por valor crea una copia de la variable pasada como argumento dentro de la función.

Al devolver como retorno una variable no modificamos realmente la variable que nos han pasado, pudiéndose almacenar en otra variable distinta.

Para asegurarnos de que el valor de la variable que nos pasan se mantiene tras la llamada a la función debemos pasarla por referencia.

La respuesta correcta es: Pasar la variable por referencia.

**Pregunta 38**

Correcta

Puntúa 1,00  
sobre 1,00

Teniendo un programa en el que queremos inicializar la variable matrix con ceros en todas sus posiciones, ¿qué fragmento de código realiza la inicialización correctamente?:

```
int main() {  
  
    int matrix[3][3];  
    int i, j;  
  
    // Código para inicializar la matriz  
  
    return 0;  
}
```

Seleccione una:

☐

```
for(i = 0; i < 3; i++){  
    for(j = 0; j < i; j++){  
        matrix[i][j] = 0;  
    }  
}
```

☐

```
for(i = 0; i < 3; i++){  
    matrix[i][0] = 0;  
}  
  
for(j = 0; j < 3; j++){  
    matrix[0][j] = 0;  
}
```

☒

```
for(i = 0; i < 3; i++){  
    for(j = 0; j < 3; j++){  
        matrix[i][j] = 0;  
    }  
}
```



Respuesta correcta

```
/* Este código sólo inicializa la primera fila y la primera columna de la matriz */

for(i = 0; i < 3; i++){
    matrix[i][0] = 0;
}

for(j = 0; j < 3; j++){
    matrix[0][j] = 0;
}

/* Este código sólo inicializa algunos elementos de la matriz */
```

```
for(i = 0; i < 3; i++){
    for(j = 0; j < i; j++){
        matrix[i][j] = 0;
    }
}
```

/\* Este código sí recorre todas las posiciones de la matriz e inicializa a cero todos los elementos \*/

```
for(i = 0; i < 3; i++){
    for(j = 0; j < 3; j++){
        matrix[i][j] = 0;
    }
}
```

La respuesta correcta es:

```
for(i = 0; i < 3; i++){
    for(j = 0; j < 3; j++){
        matrix[i][j] = 0;
    }
}
```

**Pregunta 39**

Correcta

Puntúa 1,00  
sobre 1,00


Dado el siguiente programa:

```
#include <stdio.h>

int main()
{
    int nota;
    printf("Introduzca la nota del examen [0,10] y pulse intro:");
    scanf ("%d",&nota);
    switch (nota)
    {
        case 0:
        case 1:
        case 2:
        case 3:
        case 4:
            printf ("Suspenso\n");
        case 5:
            break;
        case 6:
            printf ("Aprobado\n");
        case 7:
        case 8:
            printf ("Notable\n");
        case 9:
        case 10:
            printf ("Excelente\n");
    }
    return 0;
}
```

Si introducimos por teclado la nota de 4, el programa muestra por pantalla:

Seleccione una:

- ☐ Nada, ya que no compila porque contiene errores.
- ☒ Suspenso 
- ☐ Compila pero no muestra nada por pantalla

SOLUCIÓN:

Muestra por pantalla "Suspenso" , se ejecuta hasta el break y termina el programa

La respuesta correcta es: Suspenso

**Pregunta 40**

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>

int main() {
    int i, j;

    for (i=0;i<2;i++)
        for(j=0;j<3;j++)
            printf("%d", i);

    return 0;
}
```

Seleccione una:

- ☒ 000111 ✓
- ☐ 012012
- ☐ 010101

**SOLUCIÓN:**

- La variable i se inicializa a 0.
- Como  $i < 2$  ( $0 < 2$ ) es cierto, se ejecuta el bucle interior:
  - La variable j se inicializa a 0.
  - Como  $j < 3$  ( $0 < 3$ ) es cierto, se ejecuta el cuerpo del bucle interior:
    - Se escribe i, es decir, 0.
    - Se incrementa j, que pasa a valer 1.
  - Como  $j < 3$  ( $1 < 3$ ) es cierto, se ejecuta el cuerpo del bucle interior:
    - Se escribe i, es decir, 0.
    - Se incrementa j, que pasa a valer 2.
  - Como  $j < 3$  ( $2 < 3$ ) es cierto, se ejecuta el cuerpo del bucle interior:
    - Se escribe i, es decir, 0.
    - Se incrementa j, que pasa a valer 3.
  - Como  $j < 3$  es falso, se acaba el bucle interior.
  - Se incrementa i, que pasa a valer 1.
- Como  $i < 2$  ( $1 < 2$ ) es cierto, se ejecuta el bucle interior:
  - La variable j se inicializa a 0.
  - Como  $j < 3$  ( $0 < 3$ ) es cierto, se ejecuta el cuerpo del bucle interior:
    - Se escribe i, es decir, 1.
    - Se incrementa j, que pasa a valer 1.
  - Como  $j < 3$  ( $1 < 3$ ) es cierto, se ejecuta el cuerpo del bucle interior:
    - Se escribe i, es decir, 1.
    - Se incrementa j, que pasa a valer 2.
  - Como  $j < 3$  ( $2 < 3$ ) es cierto, se ejecuta el cuerpo del bucle interior:
    - Se escribe i, es decir, 1.
    - Se incrementa j, que pasa a valer 3.
  - Como  $j < 3$  es falso, se acaba el bucle interior.
  - Se incrementa i, que pasa a valer 2.
- Como  $i < 2$  ( $2 < 2$ ) es falso, se acaba el bucle exterior.

En resumen, el programa escribe 000111.

La respuesta correcta es: 000111

**Pregunta 41**

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuál de los siguientes códigos lee correctamente un número entero que introduce el usuario y luego lo muestra por pantalla?

Seleccione una:

☐

```
int opcion;  
scanf("%d", &opcion);  
printf("El numero introducido es %d\n", &opcion);
```

☐

```
int opcion;  
scanf("%d", opcion);  
printf("El numero introducido es %d\n", opcion);
```

☒

```
int opcion;  
scanf("%d", &opcion);  
printf("El numero introducido es %d\n", opcion); ✓
```

**SOLUCIÓN:**

La variable opcion ha de llevar & en el scanf pero no en el printf.

La respuesta correcta es:

```
int opcion;  
scanf("%d", &opcion);  
printf("El numero introducido es %d\n", opcion);
```



## Pregunta 42

Correcta

Puntúa 1,00  
sobre 1,00

Si definimos la función `areaTriangulo()` como sigue:

```
double areaTriangulo (int base, int altura) {  
    double area;  
    area=base*altura/2.0;  
    return area;  
}
```

¿Cómo debemos llamarla dentro de main para que se escriba el valor devuelto en la pantalla?

```
#include <stdio.h>
```

```
int main () {  
    int ba, al;  
    double ar;  
    printf("Introduce la base del triangulo: ");  
    scanf("%d", &ba);  
    printf("Introduce la altura del triangulo: ");  
    scanf("%d", &al);  
  
    // Llamada a la función  
  
    printf("El area del triangulo de base %d y altura %d es %.2lf", ba, al, ar);  
    return 0;  
}
```

Seleccione una:

- ☐ `areaTriangulo();`
- ☒ `ar=areaTriangulo(ba, al);` ✓
- ☐ `ar=areaTriangulo();`

### EXPLICACIÓN:

La función necesita dos argumentos de tipo int para ejecutarse y devuelve un valor de tipo double. Por lo tanto, la llamada correcta es:

```
ar = areaTriangulo (ba, al);
```

pues la variable ar de main es de tipo double y las variables ba y al son ambas de tipo int.  
La respuesta correcta es: `ar=areaTriangulo(ba, al);`

**Pregunta 43**

Correcta

Puntúa 1,00  
sobre 1,00

1. ¿Qué diferencia existe entre scanf y gets?

Seleccione una:

- ☒ scanf no permite recoger diferentes cadenas con espacios mientras que gets sí ✓
- ☐ Ninguna, hacen lo mismo
- ☐ scanf permite recoger diferentes cadenas con espacios mientras que gets no

SOLUCIÓN:

scanf no es capaz de leer por teclado cadenas más allá de un espacio. Para esto se ha de utilizar la función gets.

La respuesta correcta es: scanf no permite recoger diferentes cadenas con espacios mientras que gets sí

**Pregunta 44**

Correcta

Puntúa 1,00  
sobre 1,00

Si queremos implementar un menú de forma que se solicite una opción hasta que el usuario pulse 3 (opción de salir del programa), ¿qué instrucciones deberíamos utilizar en las líneas marcadas con (1) y (2) en el siguiente código para que funcione correctamente?

```
#include <stdio.h>
#include <stdlib.h>
```

```
typedef enum {altaUsuario=1, nuevaSesion, salir} MenuInicial;
```

```
int main() {
```

```
int opcion;
```

```
//(1)
```

```
{
```

```
printf ("\n");
```

```
printf ("1. Dar de alta a una nueva persona.\n");
```

```
printf ("2. Iniciar una sesion para un usuario.\n");
```

```
printf ("3. Salir del programa.\n\n");
```

```
printf ("Elija una opcion: ");
```

```
scanf ("%d", &opcion);
```

```
switch (opcion)
```

```
{
```

```
case altaUsuario:
```

```
printf("\nAlta usuario\n\n");
```

```
break;
```

```
case nuevaSesion:
```

```
printf("\nNueva sesion.\n\n");
```

```
break;
```

```
case salir:
```

```
printf ("\nAdios.\n\n");
```

```
break;
```

```
default:
```

```
printf ("\nOpcion incorrecta.\n\n");
```

```
}
```

```
} //(2)
```

```
return 0;
```

```
}
```

Seleccione una:

- ☒ (1) do (2) while (opcion!=salir); ✓
- ☐ (1) while (opcion==salir) (2) -
- ☐ (1) do (2) while (opcion==altaUsuario || opcion==nuevaSesion);

SOLUCIÓN:

La respuesta (1) while (opcion==salir) (2) - no es correcta dado que el bucle termina cuando opcion es cualquier otra opción que no sea salir, es decir, justo el comportamiento contrario al que se busca.

La respuesta (1) do (2) while (opcion==altaUsuario || opcion==nuevaSesion); tampoco es válida ya que si el usuario introduce una opción incorrecta (que no sea altaUsuario ó nuevaSesion) el bucle termine, pero queremos que siga, y vuelva a pedir al usuario que introduzca una opción.

La respuesta correcta es: (1) do (2) while (opcion!=salir);

#### Pregunta 45

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuál de las siguientes instrucciones utilizaremos para mostrar el tamaño en bytes de una variable de tipo float?

Seleccione una:

- ☐ printf("El tamaño de una variable de tipo float es: %f", sizeof(float));
- ☐ printf("El tamaño de una variable de tipo float es: ", sizeof(float));
- ☒ printf("El tamaño de una variable de tipo float es: %d", sizeof(float)); ✓

SOLUCIÓN:

sizeof devuelve un valor de tipo entero, por lo que se ha de utilizar el formato %d

La respuesta correcta es: printf("El tamaño de una variable de tipo float es: %d", sizeof(float));

**Pregunta 46**

Correcta

Puntúa 1,00  
sobre 1,00

¿Cómo se debe invocar a la función correspondiente para que muestre el menu inicial y se recupere la opción introducida por el usuario correctamente?

```

#include <stdio.h>

typedef enum {altaUsuario = 1, nuevaSesion, salir} mInicial;

void imprimeMenu() {
    printf("Bienvenido:\n");
    printf("1.- Dar de alta a un nuevo usuario\n");
    printf("2.- Iniciar una nueva sesion\n");
    printf("3.- Salir\n");
    printf("Seleccione una opcion: ");
}

int menuInicial1() {
    int opcion = 1;
    do {
        imprimeMenu();
        scanf("%d", &opcion);
        if (opcion < altaUsuario || opcion > salir)
            printf("Opcion incorrecta\n");
    } while (opcion < altaUsuario || opcion > salir);
    return opcion;
}

void menuInicial2() {
    int opcion = 1;
    do {
        imprimeMenu();
        scanf("%d", &opcion);
        if (opcion < altaUsuario || opcion > salir)
            printf("Opcion incorrecta\n");
    } while (opcion < altaUsuario || opcion > salir);
}

void menuInicial3(int opcion) {
    do {
        imprimeMenu();
        scanf("%d", &opcion);
        if (opcion < altaUsuario || opcion > salir)
            printf("Opcion incorrecta\n");
    } while (opcion < altaUsuario || opcion > salir);
}

int main() {
    int opcion;
    //Llamada a la funcion
    switch (opcion) {
        case altaUsuario:
            printf("Has elegido dar de alta a un usuario\n");
            break;
        case nuevaSesion:
            printf("Has elegido iniciar una nueva sesion\n");
            break;
        case salir:

```



```
        printf("Adios\n");
        break;
    default:
        printf("Opcion incorrecta\n");
        break;
    }
    return 0;
}
```

Seleccione una:



menuInicial2();



menuInicial3(opcion);



opcion = menuInicial1();



#### SOLUCIÓN:

De las tres posibles funciones la correcta es "menuInicial1", ya que es la única capaz de devolver a main el valor introducido por el usuario (mediante el comando "return opcion;"). La función "menuInicial2" no es válida ya que no devuelve nada, por lo que "main" no recibirá el valor introducido por el usuario. La función "menuInicial3" es incorrecta ya que, al recibir la variable opcion como argumento (paso por valor) aunque se modifique, el ámbito de la variable es local para esta función, por lo que no llegará el valor guardado a "main".

La respuesta correcta es:

opcion = menuInicial1();

**Pregunta 47**

Correcta

Puntúa 1,00  
sobre 1,00

¿Para qué sirve la instrucción break cuando aparece en un caso de la instrucción switch?

Seleccione una:

- ☒ para salir de la instrucción switch ✓
- ☐ para saltar las instrucciones que aparezcan hasta la siguiente llave de cierre }
- ☐ para terminar la ejecución del programa

**SOLUCIÓN:**

La instrucción break cuando aparece en un caso de la instrucción switch hace que salgamos de la instrucción switch y no examinemos el resto de los casos.

La respuesta correcta es: para salir de la instrucción switch

**Pregunta 48**

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>
int main ()
{
    int i;

    for (i=2; i>0; i--) {
        printf ("%d", i);
    }
```

```
    return 0;
}
```

Seleccione una:

- ☐ 210
- ☒ 21 ✓
- ☐ 10

**SOLUCIÓN:**

- La variable i se inicializa a 2.
- Como  $2 > 0$  es cierto, se ejecuta el cuerpo del bucle:
  - Se escribe i, es decir, 2.
  - Se decrementa i, que pasa a valer 1.
- Como  $1 > 0$  es cierto, se ejecuta el cuerpo del bucle otra vez:
  - Se escribe el valor de i, es decir, 1.
  - Se decrementa i, que pasa a valer 0.
- Como  $0 > 0$  es falso, se termina el bucle.

Por lo tanto, el programa escribe 21.

La respuesta correcta es: 21


**Pregunta 49**

Correcta

Puntúa 1,00  
sobre 1,00

¿Cuál de los siguientes **NO** es un operador lógico?

Seleccione una:

- ☐ &&
- ☒ = 
- ☐ ||

Respuesta correcta

SOLUCIÓN:

El operador = no es un operador lógico. Es el operador de asignación

La respuesta correcta es: =

**Pregunta 50**

Correcta

Puntúa 1,00  
sobre 1,00

¿Qué muestra el siguiente programa si el usuario introduce un 4?

#include &lt;stdio.h&gt;

```
typedef enum {
    altaUsuario = 1, nuevaSesion, salir
} Opcion;

int main()
{
    int opcion;
    do
    {
        printf("Bienvenido\n");
        printf("1.- Dar de alta un usuario.\n");
        printf("2.- Iniciar una nueva sesion.\n");
        printf("3.- Salir.\n");
        printf("Selecciona una opcion: ");
        scanf("%d", &opcion);
        switch (opcion)
        {
            case altaUsuario:
                printf("\nHas seleccionado dar de alta a un usuario\n");
                break;
            case nuevaSesion:
                printf("\nHas seleccionado iniciar una nueva sesion\n");
                break;
            case salir:
                printf("\nAdios\n");
            default:
                printf("\nOpcion incorrecta\n");
        }
    } while(opcion==altaUsuario || opcion== nuevaSesion);

    return 0;
}
```

Seleccione una:

- ☐ Nada, puesto que no entra en el bucle, y después termina
- ☐ El mensaje "Opcion incorrecta" y vuelve a mostrar el menu
- ☒ El mensaje "Opcion incorrecta" y después termina ✓

**SOLUCIÓN:**

Al pulsar la tecla "4", el switch entra en el "default", mostrando el mensaje correspondiente. Tras esto, al comprobar la condición del do{...}while(...) no se cumple y el programa termina. La respuesta correcta es: El mensaje "Opcion incorrecta" y después termina

[Volver a: General ➞](#)