

PROGRAMACIÓN I

Comenzado el domingo, 18 de noviembre de 2018, 11:16

Estado Finalizado

Finalizado en domingo, 18 de noviembre de 2018, 11:57

Tiempo empleado 40 minutos 41 segundos

Puntos 41,00/50,00

Calificación 8,20 de 10,00 (82%)

Pregunta 1

Correcta

Puntúa 1,00
sobre 1,00

¿Qué escribe el siguiente programa?

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

```
int main ()  
{
```

```
    int i=0, x=4;
```

```
    if (x<5)  
        while (x>2)
```

```
    {  
        i++;  
        x--;
```

```
    }  
    else  
    do
```

```
    {  
        i+=2;  
        x++;  
    }while (x%2);
```

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

```
    return 0;  
}
```

Seleccione una:

☒ 2 ✓

☐ 0

☐ 1

SOLUCIÓN:

El programa cumple la condición del if pues x es menor que 5.

El bucle empieza pues 4 es mayor que 2, incrementa i a 1 y decrementa x a 3.

El bucle se repite pues 3 es mayor que 2, incrementa i a 2 y decrementa x a 2.

El bucle no continúa y el programa muestra 2 por pantalla.

La respuesta correcta es: 2



Pregunta 2

Incorrecta

Puntuá -0,50
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>

int main () {
    int k,l,n;
    k=n=0;
    while (k<100) {
        l=0;
        while (l<100){
            n++;
            l++;
        }
        k++;
    }
    printf("%d\n",n);
```

```
    return 0;
}
```

Seleccione una:

- ☐ 200
- ☐ 10000
- ☒ 100 ✖

SOLUCIÓN:

- El bucle exterior se ejecuta 100 veces.
 - Para cada ejecución del bucle exterior, el interior se ejecuta 100 veces.
 - En el cuerpo del bucle interior se incrementa la variable n.

Por lo tanto, n se incrementa 100 por 100 veces, es decir, 10000 veces.

La respuesta correcta es: 10000

Pregunta 3

Correcta

Puntúa 1,00
sobre 1,00

¿Qué muestra por pantalla el siguiente programa?

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

```
void funcion1(char letrax, char *letray)
{
    letrax++;
    (*letray)++;
}
```

```
int main()
{
    char car='b';
    funcion1(car, &car);
    printf("%c %c", car, car);
    return 0;
}
```

Seleccione una:

- ☐ c d
- ☒ c c ✓
- ☐ d d

EXPLICACIÓN:

La variable car se pasa por valor y por referencia a la función funcion1.

Sólo la variable que recibe el valor por referencia (letray) hace afecta al valor de car una vez se ha salido de la función.

Por lo tanto car se aumenta en uno y pasa a valer 'c'.

La respuesta correcta es: c c

Pregunta 4

Correcta

Puntúa 1,00
sobre 1,00

¿Qué muestra por pantalla el siguiente programa?

```
#include <stdio.h>
void f2 (int a, int *b)
{
    a++;
    *b=*b+1;
    a=a+a;
    *b=*b+*b;
    printf(" %d %d ",a,*b);
}
int main()
{
    int x=0,y=0;
    f2(x,&y);
    printf(" %d %d\n",x,y);
    return 0;
}
```

Seleccione una:

- ☐ 2 2 0 0
- ☐ 2 1 0 1
- ☒ 2 2 0 2 ✓

EXPLICACIÓN:

Dentro de la función a pasa tomar el valor de x y b el valor de *y.
Ambas se incrementan dos veces por lo que dentro de la función se imprime por pantalla 2 2.
Al salir de la función sólo se tiene en cuenta la modificación de la variable y (que es la única que se ha pasado por referencia) por lo que se muestra 0 2 por pantalla.

La respuesta correcta es: 2 2 0 2

Pregunta 5

Correcta

Puntúa 1,00
sobre 1,00

¿Qué escribe el siguiente programa?

```
#include <stdio.h>
void g (int x)
{
    x=0;
}
void f (int * p)
{
    *p=1;
    g (*p);
}
int main() {
    int x=2;

    f (&x);

    printf("%d", x);

    return 0;
}
```

Seleccione una:

- ☐ 2
- ☒ 1 ✓
- ☐ 0

EXPLICACIÓN:

La función f altera el valor de la variable x de main, que pasa a valer 1 en la instrucción *p=1. Sin embargo, la función g solo cambia el valor de su variable local x, no de la variable local de f.

La respuesta correcta es: 1

Pregunta 6

Correcta

Puntúa 1,00
sobre 1,00

Dado el siguiente código en C

```
int a=1,b=2;  
float c;
```

```
c=a/b;  
printf("%f",c);
```

¿Cual será la salida por pantalla?

Seleccione una:

- ☐ 0.5
- ☒ 0.0 ✓
- ☐ 1.0

SOLUCIÓN:

Como tanto a como b son dos variables de tipo int la división se realiza sin decimales.

La respuesta correcta es: 0.0

Pregunta 7

Correcta

Puntúa 1,00
sobre 1,00

El siguiente programa simula un control de acceso mediante usuario y contraseña. ¿Cuál de los siguientes fragmentos completa el programa para que funcione correctamente (Deja acceder sólo si el usuario y la password coinciden. En caso contrario informa y termina)?

```
#include <stdio.h>
#include <string.h>
#define TAM 100
```

```
typedef struct {
    char usuario[TAM];
    char password[TAM];
} Credencial;
```

```
int main() {
```

```
    Credencial original = {"temporal", "unasemana"}; //Credencial guardada en el
    sistema
    Credencial test; // Credencial de prueba
```

```
    printf("Introduzca el nombre de usuario: ");
    scanf("%s", test.usuario);
    printf("Introduzca la password: ");
    scanf("%s", test.password);
```

```
    // Comprobar el acceso e informar
```

```
}
```

Seleccione una:



```
if (strcmp(original.usuario, test.usuario) == 0) {
    if (strcmp(original.password, test.password) == 0) {
        printf("Bienvenido usuario: %s", test.usuario);
    } else {
        printf("Password no valida");
    }
} else {
    printf("Usuario \"%s\" no encontrado", test.usuario);
}
```



```

if (strcmp(original.usuario, test.usuario) == 0) {
    if (strcmp(original.password, test.password) == 0) {
        printf("Bienvenido usuario: %s", test.usuario);
    } else (strcmp(original.password, test.password) != 0){
        printf("Password no valida");
    }
} else (strcmp(original.usuario, test.usuario) != 0) {
    printf("Usuario \"%s\" no encontrado", test.usuario);
}

```

```

if (strcmp(original.usuario, test.usuario) == 0) {
    printf("Bienvenido usuario: %s", test.usuario);
} else {
    printf("Usuario \"%s\" no encontrado", test.usuario);
}
if (strcmp(original.password, test.password) == 0) {
    printf("Bienvenido usuario: %s", test.usuario);
} else {
    printf("Password no valida");
}

```

Respuesta correcta

SOLUCIÓN:

La respuesta correcta es:

```

if (strcmp(original.usuario, test.usuario) == 0) {
    if (strcmp(original.password, test.password) == 0) {
        printf("Bienvenido usuario: %s", test.usuario);
    } else {
        printf("Password no valida");
    }
} else {
    printf("Usuario \"%s\" no encontrado", test.usuario);
}

```

Primero se debe comprobar el nombre del usuario mediante la función strcmp. En caso de que esta devuelva 0 (cadenas iguales) comprobamos la password de la misma forma. En los casos en los que strcmp devuelve un valor distinto de 0, las cadenas no coinciden y, por lo tanto, hay que informar que el acceso no es correcto.

Las comprobaciones de usuario y password deben ir anidadas, ya que ambas tienen que darse para que el acceso sea correcto. No se puede informar de un acceso correcto si no se han comprobado antes ambos datos. Finalmente, se debe tener en cuenta que la sentencia "else" no lleva condición.

La respuesta correcta es:

^


```
if (strcmp(original.usuario, test.usuario) == 0) {  
    if (strcmp(original.password, test.password) == 0) {  
        printf("Bienvenido usuario: %s", test.usuario);  
    } else {  
        printf("Password no valida");  
    }  
} else {  
    printf("Usuario \"%s\" no encontrado", test.usuario);  
}
```

Pregunta 8

Correcta

Puntúa 1,00
sobre 1,00

Dada la siguiente definición de una cadena de caracteres

char fecha[DIM];

¿Cuál sería el tamaño mínimo del define DIM para almacenar una fecha en formato
yyyymmdd, como por ejemplo 20140912?

Seleccione una:

- ☐ 8
- ☐ 10
- ☒ 9 ✓

Respuesta correcta

El tamaño de una cadena debe de incluir el carácter fin de cadena 0 por tanto es uno más del
tamaño necesario

La respuesta correcta es: 9

Pregunta 9

Correcta

Puntúa 1,00
sobre 1,00

¿Cuántas veces se ejecuta el bucle while que aparece en el siguiente programa?

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

```
int main()  
{
```

```
    int i = 5;
```

```
    while (i < 12)
```

```
    {  
        i++;
```

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

```
    }  
    return 0;
```

```
}
```

Seleccione una:

☐ 8

☒ 7 ✓

☐ 6

SOLUCIÓN:

El bucle se ejecuta 7 veces para los valores de i = 5, 6, 7, 8, 9, 10 y 11

La respuesta correcta es: 7

Pregunta 10

Correcta

Puntúa 1,00
sobre 1,00

Dado el siguiente programa:

```
#include <stdio.h>

int main() {
    int i,num;
    double fa;
    printf("\nCalculo el factorial de: ");
    scanf("%d",&num);
    fa=1;
    // Código para calcular el factorial
    /* solucion */
    printf("\nEl factorial de %d es %lf\n",num,fa);
    return 0;
}
```

¿Cuál de las siguientes es la solución para el cálculo del factorial?

Seleccione una:

- ☐ for(i=1;i<=num;i++) fa**;
- ☒ for(i=1;i<=num;i++) fa=fa*i; ✓
- ☐ fa=fa*i;

SOLUCIÓN:

De acuerdo con la ecuación que define el factorial se puede realizar con un bucle for, y en cada iteración guardar el valor anterior que tiene volver a ser multiplicado.

La respuesta correcta es: for(i=1;i<=num;i++) fa=fa*i;

Pregunta 11

Correcta

Puntúa 1,00
sobre 1,00

Completa el siguiente programa para que calcule el factorial del número que introduce el usuario y lo muestre por pantalla:

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

```
int main() {
```

```
int numero;  
long int factorial;  
int aux;  
  
printf("Introduzca un numero: ");  
scanf("%d", &numero);  
  
// Aqui iria el codigo para calcular el factorial y mostrar por pantalla  
// el resultado
```

```
}
```

Seleccione una:



```
factorial=1;  
aux= numero;  
while(aux>0){  
    factorial *= aux;  
    aux--;  
}  
printf("El factorial de %d es %ld",numero, factorial );
```



```
factorial=1;  
while(aux>0){  
    factorial *= numero;  
    aux--;  
}  
printf("El factorial de %d es %ld",numero, factorial );
```



```
factorial=1;  
aux= numero;  
while(factorial>0){  
    aux*= numero;  
    factorial--;  
}  
printf("El factorial de %d es %ld",numero, factorial );
```

Respuesta correcta

SOLUCIÓN:

Para calcular el factorial de "numero", se debe multiplicar esa cifra, por todas sus inferiores. Así, el factorial de 5 sería:

$$5! = 5*4*3*2*1 = 120$$

Una opción para implementarlo es ir multiplicando los productos por una variable que va de "numero" hasta 1, decrementándose en cada iteración. Así, el código sería:

```
factorial=1; // Inicializamos la variable resultado
aux= numero; //Copiamos el numero a una variable auxiliar
while(aux>0){ // Iteramos mientras la variable auxiliar sea mayor que 0
    factorial *= aux; // Guardamos el producto parcial por la variable
    aux--; //Decrementamos la variable auxiliar
}
printf("El factorial de %d es %ld",numero, factorial ); //Mostramos el resultado
```

La respuesta correcta es:

```
factorial=1;
aux= numero;
while(aux>0){
    factorial *= aux;
    aux--;
}
printf("El factorial de %d es %ld",numero, factorial );
```

Pregunta 12

Correcta

Puntúa 1,00
sobre 1,00

¿Qué escribe el siguiente programa si se teclea 7?

```
#include <stdio.h>

int main () {
    int x;

    scanf ("%d", &x);

    if (x<3 || x>=1)
        printf ("1");
    else if (x<8)
        printf ("2");
    else
        printf ("3");

    return 0;
}
```

Seleccione una:

- ☒ 1 ✓
- ☐ 2
- ☐ 3

SOLUCIÓN:

Se cumple la primera condición pues x es mayor o igual que 1, por lo que muestra un 1.
La respuesta correcta es: 1

Pregunta 13

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.y 
- ☐ mi.r.s.a
- ☐ mi.x.b.c

SOLUCIÓN:

mi es de tipo U

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

En a, dentro de la estructura M, se puede referenciar al miembro c, que es de tipo V

En c, dentro de la estructura V, se puede referenciar al miembro y.

La respuesta correcta es: mi.a.c.y

Pregunta 14

Correcta

Puntúa 1,00
sobre 1,00

En un programa en C se pretende leer una fecha separada por blancos, de acuerdo con este ejemplo de ejecución:

```
Por favor, introduce una fecha: 11 3 1972
La fecha leida es 11/3/1972
```

Suponiendo que para almacenar la fecha se dispone de una variable llamada fecha de tipo Fecha definido así:

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

¿Cuál de las siguientes sentencias permitiría leer la fecha introducida de forma correcta

Seleccione una:

☐

```
scanf("%d %d %d", fecha.&dia, fecha.&mes, fecha.&anno);
```

☐

```
scanf("%d %d %d", &fecha.&dia, &fecha.&mes, &fecha.&anno);
```

☒

```
scanf("%d %d %d", &fecha.dia, &fecha.mes, &fecha.anno);
```



Respuesta correcta

La precedencia del operador "." es mayor que la del operador "&".

La respuesta correcta es:

```
scanf("%d %d %d", &fecha.dia, &fecha.mes, &fecha.anno);
```


Pregunta 15

Correcta

Puntúa 1,00
sobre 1,00

¿Qué escribe el siguiente programa?

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

```
char f (int a, char c) {  
    return c-a;  
}
```

```
int main () {  
    char c='i';  
    printf ("%c", f(3, c));
```

```
    return 0;  
}
```

Seleccione una:

- ☒ f ✓
- ☐ e
- ☐ g

EXPLICACIÓN:

La llamada f(3,c) devuelve el carácter 'i'-3, es decir, el carácter 3 posiciones a la izquierda de 'i' en el código ASCII, es decir, la f. Esto es lo que escribe la instrucción printf ().

La respuesta correcta es: f

Pregunta 16

Correcta

Puntúa 1,00
sobre 1,00

¿Cómo completarías la función volumenEsfera del siguiente programa para que calculase el volumen de una esfera de radio r ?

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

```
#define PI 3.141592
```

```
double volumenEsfera(double r) {
```

```
// Aquí iría el código que calcula el volumen de la esfera
```

```
}
```

```
int main() {
```

```
double radio = -1, volumen = 0;
```

```
do {
```

```
printf("Introduzca el radio de la esfera: ");
```

```
scanf("%lf", &radio);
```

```
if (radio <= 0)
```

```
printf("El radio debe ser mayor que 0\n");
```

```
}while (radio <= 0);
```

```
volumen = volumenEsfera(radio);
```

```
printf("El volumen de una esfera de radio %lf es %lf\n", radio, volumen);
```

```
return 0;
```

```
}
```

(NOTA: Recuerda que el volumen de una esfera es: $V = \frac{4}{3} \pi r^3$)

Seleccione una:



```
double vol;
```

```
    vol=4*(r*3)*PI/3;  
    return vol;
```



```
return 4*r^3*PI/3;
```



```
double vol;
```

```
    vol=4.0*(r*r*r)*PI/3.0;  
    return vol;
```



SOLUCIÓN:

La única forma posible de elevar al cubo de las tres formas propuestas es $r*r*r$.

La respuesta correcta es:

```
double vol;
```

```
    vol=4.0*(r*r*r)*PI/3.0;  
    return vol;
```

Pregunta 17

Incorrecta

Puntúa -0,50
sobre 1,00

Dado el siguiente programa

```
int main()
{
    int a=1;
    for (;!a;) {
        a++;
        if (a==10)
            break;
    }
    printf("%d\n",a);
    return 0;
}
```

¿Qué se escribe por pantalla?

Seleccione una:

- ☐ 11
- ☒ 10 ✖
- ☐ 1

EXPLICACIÓN:

El programa no entra en el bucle for ya que !a es equivalente a la comprobación a==0

La respuesta correcta es: 1

Pregunta 18

Correcta

Puntúa 1,00
sobre 1,00

¿Cuál de los siguientes es un dato de tipo "char"?

Seleccione una:

- ☒ 'b' ✔
- ☐ 26
- ☐ 1.5608

SOLUCIÓN:

Los caracteres son variables de tipo char

La respuesta correcta es: 'b'

Pregunta 19

Correcta

Puntúa 1,00
sobre 1,00

¿Qué hace el siguiente programa en C?

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

```
#define NUM 50
```

```
int main() {  
  
    int i = 0;  
  
    while ( i < NUM){  
        printf("%d\n", i);  
        i += 2;  
    }  
  
    return 0;  
}
```

Seleccione una:

- ☐ Escribe todos los números pares hasta 50, inclusive
- ☒ Escribe todos los números pares menores de 50 ✓
- ☐ Escribe todos los números del 0 al 50

Respuesta correcta

Al inicial el bucle con $i = 0$ e imprimir su valor, aumentándolo de dos en dos mientras que sea menor que NUM, que es una macro que representa el valor 50, obtendremos una lista con todos los números pares menores que 50.

La respuesta correcta es: Escribe todos los números pares menores de 50

Pregunta 20

Correcta

Puntúa 1,00
sobre 1,00

Dadas las siguientes declaraciones

```
int i;  
double x;  
float z;  
long u;
```

¿Cuáles de las siguientes sentencias tiene errores?

1. `scanf("%d",&x);`
2. `printf("%f\n",x);`
3. `scanf("%ld",&z);`

Seleccione una:

- ☐ la 1 y la 3
- ☒ todas ✓
- ☐ la 2

La respuesta correcta es: todas

Pregunta 21

Incorrecta

Puntúa -0,50
sobre 1,00

¿Cuál de los siguientes códigos muestra por pantalla *Ana ha nacido el dia 27* si el usuario introduce primero *Ana* y luego *27*?

Seleccione una:



```
#define DIM 100;
int main{
    int dia; char nombre[DIM];
    printf("Introduzca su nombre: ");
    scanf("%s", nombre);
    printf("Introduzca su dia de nacimiento: ");
    scanf("%d", &dia);
    printf("%s ha nacido el dia %d", nombre, dia);
    return 0;
}
```



```
#define DIM 100;
int main{
    int dia; char nombre[DIM];
    printf("Introduzca su nombre: ");
    scanf("%s", &nombre);
    printf("Introduzca su dia de nacimiento: ");
    scanf("%d", dia);
    printf("%s ha nacido el dia %d", nombre, dia);
    return 0;
}
```



```
#define DIM 100;
int main{
    int dia; char nombre[DIM];
    printf("Introduzca su nombre: ");
    scanf("%s", &nombre);
    printf("Introduzca su dia de nacimiento: ");
    scanf("%d", &dia);
    printf("%s ha nacido el dia %d", nombre, dia);
    return 0;
}
```

**SOLUCIÓN:**

Sólo en la respuesta correcta se utiliza correctamente el símbolo & en los scanf: se ha de emplear para variables de tipo int (formato %d) pero no para cadenas de caracteres (formato %s)

La respuesta correcta es:



```
#define DIM 100;
int main{
    int dia; char nombre[DIM];
    printf("Introduzca su nombre: ");
    scanf("%s", nombre);
    printf("Introduzca su dia de nacimiento: ");
    scanf("%d", &dia);
    printf("%s ha nacido el dia %d", nombre, dia);
    return 0;
}
```

Pregunta 22

Correcta

Puntuá 1,00
sobre 1,00

Con los siguientes tipos de datos:

```
typedef struct{
```

```
    int numElementos;
    int elementos[MAX_ELEMENTOS];
```

```
} Vector;
```

```
typedef struct{
```

```
    int numFilas;
    Vector FilasVectores[MAX_FILAS];
```

```
} Matriz;
```

declaramos la variable

Matriz m;

¿Cómo debemos rellenarla para que contenga 2 filas y cada fila los datos:

1 2

4 5?

Seleccione una:

- ☒ m.numFilas=2; m.FilasVectores[0].numElementos=2;
m.FilasVectores[0].elementos[0]=1; m.FilasVectores[0].elementos[1]=2;
m.FilasVectores[1].numElementos=2; m.FilasVectores[1].elementos[0]=4;
m.FilasVectores[1].elementos[1]=5; ✓
- ☐ m.numFilas=2; m.FilasVectores.numElementos=2;
m.FilasVectores.elementos={1,2,4,5};
- ☐ m.numFilas=2; m.FilasVectores.numElementos=4;
m.FilasVectores.elementos[0][0]=1; m.FilasVectores.elementos[0][1]=2;
m.FilasVectores.elementos[1][0]=4; m.FilasVectores.elementos[1][1]=5;

SOLUCIÓN:

FilasVectores es una tabla, por lo tanto se ha de especificar a qué elemento de la tabla queremos acceder antes de acceder a cualquiera de sus miembros.

La respuesta correcta es: m.numFilas=2; m.FilasVectores[0].numElementos=2;
m.FilasVectores[0].elementos[0]=1; m.FilasVectores[0].elementos[1]=2;
m.FilasVectores[1].numElementos=2; m.FilasVectores[1].elementos[0]=4;
m.FilasVectores[1].elementos[1]=5;

Pregunta 23

Correcta

Puntúa 1,00
sobre 1,00

Hemos declarado la variable **mes** como entero. ¿Qué instrucción necesitaremos para leer de teclado un valor para dicha variable?

Seleccione una:

- ☐ scanf("%s", mes);
- ☒ scanf("%d", &mes); ✓
- ☐ scanf("%d", mes);

Respuesta correcta

Como mes es una variable entera, debemos usar el formato %d y anteponer el símbolo & al nombre de la variable.

La respuesta correcta es: scanf("%d", &mes);

Pregunta 24

Correcta

Puntúa 1,00
sobre 1,00

Si tenemos definidos:

```
#define DIM 30  
#define U 4
```

```
typedef struct {
```

```
    int dia;  
    int mes;  
    int anyo;
```

```
} Fecha;
```

```
typedef struct {
```

```
    char nombre [DIM];  
    char contra [DIM];  
    Fecha fecha;
```

```
} Usuario;
```

```
typedef struct {
```

```
    int numUsuarios;  
    Usuario usuario [U];
```

```
} Sesion;
```

Y queremos que el siguiente bucle finalice cuando encuentre un usuario registrado con el mismo nombre que el usuario introducido, ¿dónde debemos colocar la instrucción *break*?

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

```
int main() {
```

```
Usuario usuario;  
Sesion sesion;  
int i;
```

```
sesion.numUsuarios = 0;  
printf("Introduzca nombre: ");  
scanf("%s", usuario.nombre);  
printf("Introduzca contrasena: ");  
scanf("%s", usuario.contra);  
printf("Introduzca dia de nacimiento: ");  
scanf("%d", &usuario.fecha.dia);  
printf("Introduzca mes de nacimiento: ");  
scanf("%d", &usuario.fecha.mes);  
printf("Introduzca anyo de nacimiento: ");  
scanf("%d", &usuario.fecha.anyo);
```

```
for (i = 0; i < sesion.numUsuarios; i++)  
{
```

```
    //(1)  
    if (strcmp(sesion.usuario[i].nombre, usuario.nombre) == 0) {  
        //(2)  
    }
```

```
    //(3)
```

```
}
```

```
if (i==sesion.numUsuarios) {  
    printf("El usuario %s se ha registrado correctamente.\n", usuario.nombre);  
    sesion.usuario[sesion.numUsuarios] = usuario;  
    sesion.numUsuarios++;  
}
```

```
else
```

```
    printf("El usuario %s ya se encuentra en la sesion.\n", usuario.nombre);
```

```
return 0;
```

```
}
```

Seleccione una:

☐ (3)

- ☐ (1)
- ☒ (2) ✓

La respuesta correcta es: (2)

Pregunta 25

Correcta

Puntúa 1,00
sobre 1,00

¿Qué muestra por pantalla el siguiente programa?

```
int main()
{
    char primera='C', segunda='A';

    primera -= segunda;

    printf("%d", primera);
    return 0;
}
```

Seleccione una:

- ☒ 2 ✓
- ☐ A
- ☐ C

SOLUCIÓN:

La instrucción `primera -= segunda;` equivale a `primera = primera - segunda;`

Por lo tanto sería `primera = 'C' - 'A';`

Restando los valores ASCII se obtendría el valor 2.

La respuesta correcta es: 2

Pregunta 26

Incorrecta

Puntuaje -0,50
sobre 1,00

Si ejecutamos el siguiente programa, ¿qué se mostrará por pantalla?

```
#include <stdio.h>

int main() {
    int x = 3, y = 4, z = 6;

    if (!x && y < 3)
    {
        printf("1\n");
    }
    else if (x || y < 3)
    {
        if (z = 7)
        {
            printf("2\n");
        }
        printf("3\n");
    }
    return 0;
}
```

Seleccione una:

☐

1

☐

2

3

☒

3



Respuesta incorrecta.

SOLUCIÓN:

Teniendo en cuenta lo siguiente:

!x es 0.

y < 3 es 0.

x es verdadero ya que x es distinto de 0.

z = 7 es una asignación (**no una comparación**) que da un valor distinto de 0, por lo que se considera también verdadera.

Por lo tanto, la condición del primer if no se cumple y la del else if y el segundo if sí se cumple.

La respuesta correcta es:



2

3

Pregunta 27

Correcta

Puntúa 1,00
sobre 1,00

En el siguiente programa en C:

```
#define MESESA 32
#define MAX_MSJ 32
```

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

```
int main()
{
    Fecha fecha;
    int diasMes[MESESA]={31,29,31,30,31,30,31,31,30,31,30,31};
    int fechaOk = 0;

    /***** Codigo para obtener una fecha adecuada del usuario *****/

    printf("\nLa fecha es correcta\n\n");

    return 0;
}
```

Se considera utilizar una de las siguientes alternativas para conseguir una fecha adecuada del usuario :

Alternativa A

```
do{
    printf("Fecha (dd mm aa): ");
    scanf("%d %d %d", &fecha.dia, &fecha.mes, &fecha.anno);
    if ((0 > fecha.anno) ||
        (0 >= fecha.mes) || (MESESA < fecha.mes) ||
        (0 >= fecha.dia) || (diasMes[fecha.mes-1]<fecha.dia))
        printf("Fecha erronea\n");
    else fechaOk = 1;
} while (!fechaOk);
```

Alternativa B

```
while (!fechaOk){
    printf("Fecha (dd mm aa): ");
    scanf("%d %d %d", &fecha.dia, &fecha.mes, &fecha.anno);
    if ((0 > fecha.anno) ||
        (0 >= fecha.mes) || (MESESA < fecha.mes) ||
        (0 >= fecha.dia) || (diasMes[fecha.mes-1]<fecha.dia))
        printf("Fecha erronea\n");
    else fechaOk = 1;
};
```

¿Cuál de las siguientes afirmaciones es correcta?

Seleccione una:

- ☐ La alternativa B es correcta pero no la A
- ☐ La alternativa A es correcta pero no la B
- ☒ Las dos alternativas son correctas ✓

Respuesta correcta

Como la variable *fechaOk* se ha inicializado a 0 en la declaración las dos alternativas son correctas

La respuesta correcta es: Las dos alternativas son correctas

Pregunta 28

Correcta

Puntúa 1,00
sobre 1,00

¿Cuántas veces se ejecuta el cuerpo del bucle del siguiente programa?

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

```
void incremento(int i){
    i++;
}
```

```
int main(){

    int i;

    for(i = 0; i < 50; i++){
        incremento(i);
    }
    return 0;
}
```

Seleccione una:

- ☒ 50 ✓
- ☐ Se trata de un bucle infinito
- ☐ 25

Respuesta correcta

En la función *incremento*, el ámbito de la variable *i* es local, por lo que aunque se realice un incremento, no afecta al índice del bucle que la llama. Por tanto, el bucle tendrá 50 iteraciones

La respuesta correcta es: 50

^

Pregunta 29

Correcta

Puntúa 1,00
sobre 1,00

¿Qué muestra por pantalla el siguiente código?

```
#include <stdio.h>

int main(){
int a; b; c;
a = 8;
b = 4;
c = 2;
printf("a=%d, b=%d, c=%d", c,b,a);
return 0;
}
```

Seleccione una:

- ☐ 8, 4, 2
- ☐ 2, 4, 8
- ☒ El programa no compila ✓

SOLUCIÓN:

En una declaración de variables los nombres de las variables han de estar separados por coma, no por punto y coma.

La respuesta correcta es: El programa no compila

Pregunta 30

Correcta

Puntúa 1,00
sobre 1,00

¿Qué escribe el siguiente programa si se teclea 8?

```
#include <stdio.h>

int main () {
int x;

scanf ("%d", &x);

if (x%2)
printf ("1");
else if (x%3)
printf ("2");
else
printf ("3");

return 0;
}
```

Seleccione una:

- ☐ 3
- ☐ 1
- ☒ 2 ✓

SOLUCIÓN:

Se cumple la segunda condición pues $x\%3$ equivale a $x\%3 \neq 0$, por lo que muestra un 2.

La respuesta correcta es: 2

Pregunta 31

Correcta

Puntúa 1,00
sobre 1,00

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

```
int f1(int i, int j) {
```

```
    return i-j;
```

```
}
```

```
int f2(int i, int j) {
```

```
    return i*j;
```

```
}
```

```
int main() {
```

```
    printf("%d", f2(f1(4,2),f2(4,2)));
```

```
    return 0;
```

```
}
```

Seleccione una:

☐ 4☒ 16 ✓☐ 0**EXPLICACIÓN:**

$f1(4,2)=2$ y $f2(4,2)=8$.

Por lo tanto, $f2(f1(4,2), f2(4,2)) = f2(2, 8) = 16$.

La respuesta correcta es: 16

Pregunta 32

Correcta

Puntúa 1,00
sobre 1,00

¿Qué imprimiría el siguiente código por pantalla?

```
#include<stdio.h>

int main(){
printf("%%%\n%%");
return 0;
}
```

Seleccione una:

- ☐ Un línea con cuatro símbolos %
- ☒ Dos líneas, cada una con dos símbolos % ✓
- ☐ Dos líneas, cada una con cuatro símbolos %

SOLUCIÓN:

Por cada dos % en un printf, se imprime sólo uno por pantalla

La respuesta correcta es: Dos líneas, cada una con dos símbolos %

Pregunta 33

Correcta

Puntúa 1,00
sobre 1,00

Teniendo en cuenta el siguiente programa:

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

```
#define MAX 100
```

```
typedef struct{  
    char nombre[MAX];  
    char director[MAX];  
    int fecha;  
    double puntuacion;  
} Pelicula;
```

```
typedef struct{  
    Pelicula peliculas[MAX];  
    int numero_de_peliculas;  
} Catalogo;
```

```
int main() {
```

```
Catalogo catalogo;  
catalogo.numero_de_peliculas = 0;  
  
catalogo.peliculas[catalogo.numero_de_peliculas].nombre = "El viaje de Chihiro";  
catalogo.peliculas[catalogo.numero_de_peliculas].director = "Hayao Miyazaki";  
catalogo.peliculas[catalogo.numero_de_peliculas].fecha = 2001;  
catalogo.peliculas[catalogo.numero_de_peliculas].puntuacion = 8.1;  
  
return 0;  
}
```

¿Qué haría falta hacer para que se pudieran seguir introduciendo más películas en el catálogo mediante el acceso `catalogo.peliculas[catalogo.numero_de_peliculas]`?

Seleccione una:

- ☐ a. Reiniciar el contador `catalogo.numero_de_peliculas` a 0
- ☐ b. No se puede acceder a la lista de películas más veces de esa manera
- ☒ c. Incrementar el valor de `catalogo.numero_de_peliculas` en 1 ✓

Respuesta correcta

La respuesta correcta es: Incrementar el valor de `catalogo.numero_de_peliculas` en 1

Pregunta 34


Incorrecta

Puntúa -0,50
sobre 1,00

Dado el siguiente programa, seleccione que muestra por pantalla:

```
#include <stdio.h>
void funcion1 (int * p)
{
    (*p)++;
    p++;
    (*p)--;
}
int main ()
{
    int num[6] = {10,14,23,17,15,8};
    int *p = &num[4];
    funcion1(p);
    printf ("%d", num[4]-num[5]);
    return 0;
}
```

Seleccione una:

- ☒ 8 
- ☐ 9
- ☐ 4

EXPLICACIÓN:

Tras llamar a funcion1 la tabla num quedaría de la forma {10,14,23,17,16,7}

La respuesta correcta es: 9

Pregunta 35

Correcta

Puntúa 1,00
sobre 1,00

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

```
int main() {
```

```
    int i;
```

```
    for (i=0;i<5;i++)  
    {
```

```
        printf("%d ", i);  
        i=i+2;
```

```
    }
```

```
    return 0;
```

```
}
```

¿Qué mostrará por pantalla el programa anterior?

Seleccione una:

- ☐ 1 4
- ☐ 0 1 2 3 4
- ☒ 0 3 ✓

La respuesta correcta es: 0 3

Pregunta 36

Correcta

Puntúa 1,00
sobre 1,00

Dadas las siguientes sentencias en C

```
int a=98;  
printf("%c",a);
```

y teniendo en cuenta que en la tabla de caracteres ASCII la posición 98 la ocupa el caracter b
¿Cuál será la salida por pantalla?

Seleccione una:

☐ 98b

☒ b ✓

☐ 98

SOLUCIÓN:

Al usar el formato %c se muestra el valor de la variable a como carácter

La respuesta correcta es: b

Pregunta 37

Correcta

Puntúa 1,00
sobre 1,00

Si en una función main aparecen las siguientes declaraciones

```
int *pnumero;
```

```
int num1, num2;
```

```
char *pchar;
```

```
char letra1;
```

```
num1 = 2;
```

```
num2 = 5;
```

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

¿Qué valor tiene **&pchar** tras hacer las siguientes asignaciones?

```
pchar = &letra1;
```

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

Seleccione una:

- ☐ 'a'
- ☐ 'B'
- ☒ La dirección de memoria de pchar ✓

EXPLICACIÓN:

&pchar se referirá a la dirección de memoria en la que se encuentra el puntero pchar.

La respuesta correcta es: La dirección de memoria de pchar

Pregunta 38

Correcta

Puntúa 1,00
sobre 1,00

dado el siguiente código en C:

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

#define DIM 100

typedef struct {
    char nombre[DIM];
    char apellido[DIM];
} name;

typedef struct {
    short yyyy,mm,dd;
} fecha;

typedef struct {
    name x;
    fecha f;
} persona;

typedef struct {
    persona u[DIM];
    long n;
} agenda;

int main()
{
    agenda z;
    z.n=0;

    while (z.n<DIM) {
        printf("escribe el nombre de la persona %ld\n",z.n+1);
        scanf("%s",z.u[z.n].x.nombre);
        z.n++;
    }
}
```

Indicar cual de las siguientes opciones es cierta

Seleccione una:

- ☒ El bucle se ejecutará 100 veces y el programa terminará ✓
- ☐ El bucle se ejecutará solo una vez y el programa terminará
- ☐ El bucle se ejecutara infinitas veces pues nunca se cumplirá la condición de salida

Respuesta correcta

La respuesta correcta es: El bucle se ejecutará 100 veces y el programa terminará

Pregunta 39

Correcta

Puntúa 1,00
sobre 1,00

Dada la siguiente declaración:

```
const int meses[12] = {31,28,31,30,31,30,31,31,30,31,30,31};
```

¿Cuál de las siguientes sentencias produce un error?

Seleccione una:

- ☒ `scanf("%d", &meses[3])` ✓
- ☐ `printf("%d", meses[3] + meses[4]);`
- ☐ `printf("%d", meses[3]);`

SOLUCIÓN:

meses se ha declarado como una contante (mediante el modificador const) por lo que su valor no se puede cambiar ni mediante una asignación ni mediante un scanf, como en este caso.

La respuesta correcta es: `scanf("%d", &meses[3])`

Pregunta 40

Correcta

Puntúa 1,00
sobre 1,00

¿Cuántas veces se mostrará el menú de inicio antes de terminar el programa si el usuario teclea como opción 3?

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

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

```
int main() {
```

```
int opcion;
```

```
do  
{
```

```
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");
```

```
}
```

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

```
return 0;
```

```
}
```

Seleccione una:

- ☐ Ninguna.
- ☒ Una. ✓
- ☐ Hasta que el usuario vuelva a introducir '3'.

SOLUCIÓN:

Al usuario se le muestra una sólo vez el menú, dado que la opción 3 se corresponde con `salir` (fíjate en la posición que tiene en el enumerado y el valor al que se inicializa `altaUsuario`). La condición de mantenimiento del bucle es que `opcion != salir`, así al tomar el valor `salir` termina.

La respuesta correcta es: Una.

Pregunta 41

Correcta

Puntúa 1,00
sobre 1,00

¿Qué se muestra por pantalla si el usuario introduce un 6?

```
#include <stdio.h>

typedef enum {enero=1, febrero, marzo, abril, mayo, junio, julio,
             agosto, septiembre, octubre, noviembre, diciembre} Mes;

typedef struct {
    int dia;
    int mes;
    int agno;
}Fecha;

int main() {
    Fecha f;

    printf("\nIntroduce el mes: ");
    scanf("%d", &f.mes);

    switch(f.mes){
        case enero:
        case marzo:
        case mayo:
        case julio:
        case agosto:
        case octubre:
        case diciembre:
            printf("\nEl mes %d tiene 31 dias\n", f.mes);
            break;
        case abril:
        case junio:
        case septiembre:
        case noviembre:
            printf("\nEl mes %d tiene 30 dias\n", f.mes);
        case febrero:
            printf("\nEl mes %d tiene 28 o 29 dias\n", f.mes);
            break;
        default:
            printf("Mes incorrecto\n");
            break;
    }

    return 0;
}
```

Seleccione una:

- ☐ El mes 6 tiene 30 días
- ☒ El mes 6 tiene 30 días
- ☐ El mes 6 tiene 28 o 29 días ✓
- ☐ El mes 6 tiene 31 días

SOLUCIÓN:

Al introducir un 6, se entraría en el "case" correspondiente a junio y se ejecutarían todas las instrucciones hasta encontrar un "break". Dado que éste no se encuentra hasta el "case" de febrero, se mostrarían las siguientes líneas por pantalla:

El mes 6 tiene 30 dias

El mes 6 tiene 28 o 29 dias

La respuesta correcta es:

El mes 6 tiene 30 dias

El mes 6 tiene 28 o 29 dias

Pregunta 42

Correcta

Puntúa 1,00
sobre 1,00

Si ejecutamos el siguiente fragmento de código, ¿qué se mostrará por pantalla?

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

```
int main()  
{  
    int a = 3;
```

```
    switch (a) {  
        case 1:  
        case 2:  
        case 3:  
            printf("Entro en 3\n");  
            a += 2;  
        case 4:  
            printf("Entro en 4\n");  
        case 5:  
            printf("Entro en 5\n");  
            break;  
        default:  
            printf("default\n");  
    }  
  
    return 0;  
}
```

Seleccione una:



Entro en 3
Entro en 4
Entro en 5



Entro en 3
Entro en 5
default



Entro en 3
Entro en 5



Respuesta correcta

SOLUCIÓN:

Al llegar a la sentencia del switch el valor de a es 3, por lo tanto se ejecuta directamente el case 3 que muestra por pantalla "Entro en 3" y suma dos al valor de a. Ahora, por la manera de ejecutarse el switch, la ejecución sigue secuencialmente hasta encontrar un break y en este caso lo encuentra es en el case 5, por lo que imprime "Entro en 4" y "Entro en 5" y ya, al encontrar el break, sale del switch.

La respuesta correcta es:

Entro en 3
Entro en 4
Entro en 5

Pregunta 43

Correcta

Puntúa 1,00
sobre 1,00

¿Cuántas veces se muestra "hola" por pantalla si ejecutamos el siguiente código?

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

```
int main()
{
```

```
    int contador=0;
```

```
    while (contador<5)
    {
        do{
```

```
        printf("hola\n");
        contador++;
```

```
    }while (contador<5);
```

```
}
```

```
return 0;
```

```
}
```

Seleccione una:

- ☐ 25
- ☒ 5 ✓
- ☐ 6

^

La respuesta correcta es: 5

Pregunta 44

Correcta

Puntúa 1,00
sobre 1,00

¿Qué escribe el siguiente programa?

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

```
int main ()  
{  
  
    int n=6,r;  
  
    while ( n )  
    {  
        r = n%2;  
        printf ("%d", r);  
        n /= 2;  
    }  
  
    return 0;  
}
```

Seleccione una:

- ☐ 010
- ☒ 011 ✓
- ☐ 110

SOLUCIÓN:

Entra en el bucle, r toma el valor 0 que imprime por pantalla y n el valor 3.
Vuelve a entrar en el bucle, r toma el valor 1 que imprime por pantalla y n el valor 1.
Vuelve a entrar en el bucle, r toma el valor 0 que imprime por pantalla y n el valor 0.
La respuesta correcta es: 011

Pregunta 45

Correcta

Puntúa 1,00
sobre 1,00

Si tenemos la matriz de números:

```
1 2 3  
4 5 6  
7 8 9
```

almacenada en la variable

```
int matriz[3][3];
```

¿Qué número se guarda en `matriz[1][1]`?

Seleccione una:

- ☐ 1
- ☒ 5 ✓
- ☐ 4

SOLUCIÓN:

`matriz[1][1]` corresponde con la segunda fila y segunda columna, esto es, el valor 5.
La respuesta correcta es: 5

Pregunta 46

Incorrecta

Puntúa -0,50
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:

- ☐ Ninguna de las otras
- ☒ 1 
- ☐ 0

EXPLICACIÓN:

La instrucción:

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

es igual que

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

Como los caracteres 'b' y 'c' son distintos, la función f devuelve 1.

Como 1 es distinto de 0, es cierto, se ejecuta printf ("0"), es decir, se escribe 0 en la pantalla.

La respuesta correcta es: 0

Pregunta 47

Correcta

Puntúa 1,00
sobre 1,00

Para hacer que el formato de las fechas que imprime un programa en C sea *dd/mm/aaaa*, donde *dd* es el día, *mm* el mes y *aaaa* el año, y que todas las cifras se completan con ceros a la izquierda (por ejemplo 03/05/2016). ¿Qué formato de la instrucción *printf* es el adecuado?

Seleccione una:

☐

```
printf("%20d/%20d/%40d\n", dia, mes, anio);
```

☐

```
printf("%02d/%02d/%04d\n", dia, mes, anio);
```

☐

```
printf("%d/%d/%d\n", dia, mes, anio);
```

Respuesta correcta

SOLUCIÓN:

El formato `"%02d"` imprime un número justificado con ceros hasta dos dígitos y el formato `"%04d"` hasta cuatro dígitos.

La respuesta correcta es:

```
printf("%02d/%02d/%04d\n", dia, mes, anio);
```

Pregunta 48

Correcta

Puntúa 1,00
sobre 1,00

¿Qué salida produce el siguiente programa?

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

```
int main() {
    int i, n = 77;
```

```
    for (i = 2; i < sqrt(n) && (n % i); i++) {
        printf("%d", n % i);
    }

    return 0;
}
```

Seleccione una:

- ☐ 121250
- ☐ 1212505
- ☒ 12125 ✓

Respuesta correcta

SOLUCIÓN:

El bucle imprime el resto de n módulo i, donde i va desde 2 hasta la raíz de n, siempre y cuando n no sea divisible por i. Y como podemos comprobar fácilmente:

 $77 \% 2 = 1$ $77 \% 3 = 2$ $77 \% 4 = 1$ $77 \% 5 = 2$ $77 \% 6 = 5$ $77 \% 7 = 0$

de esta manera se termina el bucle sin imprimir $77 \% 7$.

La respuesta correcta es: 12125

Pregunta 49

Correcta

Puntúa 1,00
sobre 1,00

¿Con cuál de las siguientes declaraciones es posible cambiar el valor de x con una instrucción posterior?

Seleccione una:

- ☒ `int x = 10;` ✓
- ☐ `#define x 10`
- ☐ `const int x = 10;`

^

SOLUCIÓN:

Sólo se puede modificar el valor de las variables, nunca de las macros (`#define x 10`) ni de las constantes (`const int x = 10;`)

La respuesta correcta es: `int x = 10;`

Pregunta 50

Correcta

Puntúa 1,00
sobre 1,00

¿Qué sentencia habría que poner entre los `printf` para que el usuario pudiera introducir un número de opción?

```
int main ()
{
    int opcion;
    printf ("Elija entre 1, 2, y 3: ");
    /* Falta una línea de código */
    printf ("La opcion es %d\n", opcion);
    return 0;
}
```

Seleccione una:

- ☐ `readline("%d", &opcion);`
- ☒ `scanf("%d", &opcion);` ✓
- ☐ `scanf("%d", opcion);`

SOLUCIÓN:

La instrucción `scanf("%d", &opcion);` sirve para leer un número entero y guardarlo en la variable `opcion` e incluye el `&` delante del nombre de la variable.

La respuesta correcta es: `scanf("%d", &opcion);`

Volver a: General ➡