

Programación

Test Autoevaluación Tema 5

Autores:

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



Universidad
Carlos III de Madrid
www.uc3m.es



1. ¿Cuál es el resultado del siguiente programa?

```
#include <stdio.h>

void imprimeDatos(int b, int a);

int main(void) {
    int a, b;
    a= 5;
    b = 10;
    imprimeDatos(5,10);
    return 0;
}

void imprimeDatos(int b, int a){
    printf("VALORES: %d y %d", a, b);
    return;
}
```

- VALORES: 10 y 5.
 - VALORES: 5 y 10.
 - El programa no compila porque utiliza variables globales.
 - El programa no compila porque la llamada a la función `imprimeDatos` es incorrecta.
2. Señale la forma correcta de terminar la frase: "En el programa principal, el valor de una variable que se pasa como parámetro por valor a una función"
- no puede ser modificado.
 - cambia cuando dentro de la función se modifica el valor de dicho parámetro.
 - no cambia aunque dentro de la función se modifique el valor de dicho parámetro.
 - no puede conocerse después de la llamada, puesto que la variable se destruye.
3. Dado el siguiente programa: ¿cuál de las siguientes líneas de código podría escribirse en la línea de guiones (----) para que el programa compilase sin errores?

```
#include <stdio.h>
float mi_funcion(float a, int b, char c);

int main()
{
    float num_a=5.5;
    int num_b=4.0;
    char letra;
    -----
    return(0);
}

float mi_funcion(float a, int b, char c)
{
    c='k';
    return(a+b);
}
```

- `printf ("%f", mi_funcion(num_a,num_b,letra));`
 - `printf ("%f", mi_funcion(num_b,num_a,letra));`
 - El programa no compila porque la sentencia de asignación `int num_b=4.0;` es incorrecta.
 - Cualquiera de las dos primeras
4. Dado el siguiente programa, señale la afirmación correcta.

```
#include <stdio.h>
int d (int e,int f);

int main(void)
{
    int a,b,c;
    c=d(a,b);
    printf ("%i" , c);
    return 0;
}
int d (int e,int f){
    int g,h;
    return (e+g+f+h);
}
```

- a, b, c son variables globales.
 - a, b, c son variables locales.
 - e, f son parámetros reales.
 - e y f son variables globales.
5. ¿Cuál de las siguientes afirmaciones es la más apropiada para continuar la siguiente frase: “Las funciones
- ...no son necesarias para programar y casi ningún programador las usa”.
 - ...facilitan la programación, evitando tener que repetir código que se utiliza varias veces en el mismo programa”.
 - ...siempre deben tener al menos un parámetro de entrada”.
 - ... facilitan la programación, al permitir la transformación de los datos”.
6. ¿Cuál es el resultado que se muestra por pantalla tras ejecutar el siguiente programa?

```
#include <stdio.h>
int producto (int a,int b);

int main(void){
    int n1=5, n2=7;
    n2 = producto(n1,n2);
    printf ("%i, %i", n2, n1);
    return 0;
}
int producto (int a,int b){
    a= 20;
    return (a * b);
}
```

- 140, 5
- 140, 20
- 35, 5
- 35, 20

7. Señale la respuesta **falsa** respecto a la técnica de programación modular:
- Se basa en la descomposición del programa principal en otros más simples.
 - Establece que el programa principal solamente puede contener las operaciones fundamentales.
 - Se basa en el diseño de subprogramas, que pueden reutilizarse en otros programas.
 - Permite crear un programa principal más corto y legible.

8. Señale la afirmación correcta respecto al siguiente programa

```
#include <stdio.h>
int Calcular (int a,int b,int s,int r);

int main(void)
{
    int n1=8,n2=2, suma, resta, producto,code;
    code = Calcular (n1,n2,suma,resta);
    printf ("%i, %i, %i", suma,resta,producto);
    system ("PAUSE");
    return 0;
}

int Calcular (int a,int b,int s,int r)
{
    int p;
    s =a+b;
    r=a-b;
    p=a*b;
    return (0);
}
```

- Muestra los resultados correctos para la suma y resta de n1 y n2, pero no el resultado correcto para el producto.
 - No muestra los resultados correctos ni de la suma, ni de la resta ni del producto de n1 y n2.
 - No compila, puesto que los parámetros formales y los reales no tienen el mismo nombre en la llamada y en el prototipo.
 - Utiliza variables globales en la función Calcular.
9. ¿Por qué no deben usarse variables globales dentro de una función?
- Porque sus valores solo son accesibles desde el programa main.
 - Porque dificulta la depuración y mantenimiento del código.
 - Sí se deben usar, siempre que sea necesario acceder a sus valores.
 - Porque su ámbito es el local de la función.

10. Dado el siguiente programa, señale la afirmación correcta :

```
#include <stdio.h>
int Intercambio(int*, int*);

int main(void) {
    int x,y,z;
    x=5;
    y=10;
    z=Intercambio(&x,&y);
    printf("x=%i\t y=%i\n", x, y);
    system("pause");
    return 0;
}

int Intercambio(int* a, int* b)
{
    int aux;
    aux=b;
    b=a;
    a=aux;
    return (0);
}
```

- Imprime por pantalla x=5 y=10
- Imprime por pantalla x=10 y=5
- El programa no compila porque hay un error en la declaración de la función (prototipo).
- El programa no compila porque hay un error en el cuerpo de la función.