

1. Modificar el ejemplo HolaMundo, duplicando la llamada a la función `printf`. Compilarlo y comprobar el nuevo comportamiento.
2. En el programa anterior, introduzca errores de sintaxis y observe los errores que emite el compilador.
3. Encontrar los efectos de embeber las secuencias: `\n`, `\t`, `\b`, `\r` en medio del string "Hola Mundo".
4. Qué salida espera encontrar si compila y ejecuta el siguiente programa, verifíquelo.

```
#include <stdio.h>

int main (void)
{
    printf ("Testing...");
    printf ("...1");
    printf ("...2");
    printf ("..3");
    printf ("\n");
    return 0;
}
```

5. ¿Cuál sería la salida del siguiente programa? Verifíquelo.

```
#include <stdio.h>
int main (void)
{
    char c, d;
    c = 'd';
    d = c;
    printf ("d = %c\n", d);
    return 0;
}
```

6. Escriba un programa que evalúe el polinomio $3x^3 - 5x^2 + 6$ para $x = 2.55$.
7. Para expresar el color de un pixel en una imagen es común empaquetar las intensidades de las componentes roja, verde y azul (RGB) dentro de un número entero, en donde los bits 0-7 corresponden a la componente R, los bits 8-15 a la componente G y los bits 16-23 a la componente azul.
 - a. Realice un programa que solicite las componentes RGB de un pixel (con intensidades entre 0 y 255 para cada componente) e imprima su color como un entero utilizando el empaquetamiento citado.
 - b. Realice un programa que reciba un color empaquetado e imprima las componentes RGB del pixel.
8. El formato para imprimir variables en la salida estándar a través de la función `printf` es

```
printf("Estas son variables %tipo1, %tipo2, ...",variable1,variable2,...);
```

donde **%tipo** define el formato y el tipo de variable a imprimir. Por ejemplo, para variables enteras el tipo es **%d**, mientras que para variables de punto flotante se pueden usar **%f** o **%e** (existen más formatos enteros y flotantes, pero estos son los más usuales). Además, se pueden utilizar modificadores entre el símbolo '%' y el tipo de variable para cambiar el formato de impresión. Por ejemplo, '+' fuerza la impresión del signo de una variable numérica.

a) Escriba un programa que imprima una variable entera (int) y una de punto flotante simple (float) definidas previamente. Pruébalo con números positivos y negativos y usando diferentes cantidades de cifras significativas. Pruebe además usar **%+d** o **%+f** y diga qué cambia. ¿Qué ocurre si mezcla los tipos e intenta imprimir un entero como real o viceversa?

b) Otro modificador de utilidad es el que permite definir el número de decimales a imprimir. Corra un programa que defina el número $\pi = 3.141592653589793238$ como una variable de punto flotante de precisión doble y la imprima usando los formatos **%f**, **%.2f**, **%.8f** y **%.15f**. ¿Cuál es el resultado? Si ahora define una nueva variable de precisión simple **pi_float** y le asigna el mismo valor numérico que a **pi**, ¿nota algo raro al imprimir ambas variables con 16 dígitos de precisión?

c) ¿Cómo haría para imprimir la constante de Planck ($6.62606896 \times 10^{-34}$ J s) con 7 dígitos significativos?

9. Realice un programa que solicite al usuario el ingreso de 2 números complejos (se deben ir solicitando los atributos por separado) y luego los sume, multiplique e imprima los resultados. Tenga en cuenta que los números complejos son tipos definidos por el usuario por lo que las operaciones sobre este tipo (suma, resta, multiplicación, etc.) deben ser implementadas también por el usuario.