Consejos para Programación Segura y Estable en C

1 Controlar Buffer Overflow

Poner todos lo controles necesarios para evitar acceder (lectura o escritura) fuera del espacio de memoria de vectores y matrices.

Leer cadenas de caracteres mediante la función `fgets`, en lugar de `gets` o `scanf o fscanf.`

En caso de duda sobre el tamaño de las cadenas de caracteres utilizar `strncpy`, `strncmp` y `strncat`, en lugar de `strcpy`, `strcmp` o `strcat`.

Es muy peligroso, y casi siempre incorrecto, hacer asignaciones del tipo `cadena="Hola";`

2 Medidas para evitar bucles infinitos

En bucles `while` y `do-while` comprobar que las variables que intervienen en la condición se modifican dentro del bucle. Verificar que para cualquier valor inicial, existe solución o verificar el rango de valores iniciales válidos antes de entrar en el bucle

En funciones recursivas, comprobar que las variables que intervienen en la condición de salida se modifican antes de hacer la siguiente llamada recursiva.

3 Desconfiar del valor de las variables

Hay que tener en cuenta que los valores de las variables pueden no ser válidos o estar fuera de rango, por lo tanto:

Verificar si los rangos son válidos antes de realizar operaciones críticas (teniendo en cuenta resultados intermedios de la operación): división (¿es cero?), raíz cuadrada (¿es negativo?), operaciones con enteros (¿se produce desbordamiento?).

Controlar los errores de lectura de archivos o de teclado. Todas las funciones de lectura devuelven códigos de error o proporcionan la información necesaria para detectarlos.

4 Estabilidad del programa

Inicializar variables siempre que sea necesario para evitar comportamientos imprevistos.

Llevar buen control de la memoria dinámica:

- Evitar perder memoria asignada, pues el programa agotaría la memoria del sistema.
- Garantizar que el programa nunca libera memoria no asignada (o libera memoria dos veces).

Tener en cuenta casos poco probables: el archivo no existe, el disco está lleno, no hay memoria suficiente, la conexión no responde, etc. Es decir, hay que verificar los valores retornados por las funciones para controlar los errores (imprescindible: `fopen, malloc, calloc, realloc`).