1. Ladrichuleta para micro

1. Al final del programa va la directiva `END`. Si va seguida de una etiqueta, la ejecución comienza en dicha etiqueta (T2.5, p.32)
2. Si la función se llama desde C tiene que estar declarada con `_nombre`.
3. Si el modelo de compilación de C es largo, entonces se utiliza `FAR` para procedimientos (la dirección de retorno son 4 bytes) y los punternos son de 4 bytes. Si el modelo es corto entonces se utiliza `NEAR` (la dirección de retorno son 2 bytes) y los punteros son de 2 bytes.
4. Si un procedimiento de C devuelve una palabra lo hace en `AX`. Si devuelve dos palabras, lo hace en `DX:AX`.
5. Todo procedimiento debe no tener efectos colaterales (los registros no pueden modificarse excepto en un procedimiento de C que no devuelva `void`).
6. Hay que tener en cuenta el signo de los enteros al elegir las instrucciones JMP.
7. Utilizar `CMP` en vez de `SUB` y `TEST` en vez de `AND` para las condiciones para saltar.
8. No utilizar ni `CMP` ni `SUB` si se va a comparar con `0` o con el signo y si sobre lo que se va a comparar ha sido utilizado en la última operación aritmética o lógica (las flags se mantienen para que las use la instrcción de salto).
9. Si se quiere utilizar solo un byte de memoria direccionando por registro hay que hacer un cast con `BYTE PTR` porque por defecto coge una palabra.
10. Si se accede a una variable en memoria no es necesario casting porque el tipo se define en la declaración (`VARIABLE db/dw`).
11. Si un procedimiento llama a otros y pasa argumentos por la pila haciendo push suele ser necesario hacer `add sp, N` (N entero) para limpiar la pila antes de retornar en el primer procedimiento. No se pueden utilizar `pop`s porque contaminarían los registros.
12. Los `jmp` a etiquetas que están justo debajo no hacen falta (basta con dejar que siga).
13. Comprobar los `INC`s y `DEC`s que suelen estar cambiados.
14. Al leer un `char` de la pila que hemos pasado por parámetros descartamos la parte más significativa (que deberían ser ceros y está en la dirección más baja por ser little endian).