



## Práctica C - Repaso

1. Escriba un programa que declare algunas variables locales, e imprima las direcciones de memoria de las mismas. Pruebe declarar un arreglo de caracteres, y verifique las direcciones de sus elementos son contiguas.
2. Defina una estructura **carta** para representar una carta de la baraja española (represente el palo con una enumeración). Cree un arreglo de 48 cartas, llámelo mazo y llénelo con las cartas correspondientes.
3. Implemente una función **struct carta azar(struct carta[], int)** que reciba un mazo, la longitud del mismo, y devuelva una carta al azar del mazo pasado.
4. Implemente una función **void setzerozero(int [])** que ponga en cero el primer elemento del arreglo recibido. Verifique desde la función llamante que efectivamente modifica este valor. Analice el por qué pasa esto.
5. Explique el tipo de la función **malloc**, ¿qué valor retorna la función en caso de que no pueda reservar el espacio solicitado?
6. Implemente una estructura **contacto** que tenga como campos: una cadena para el nombre de una persona, una cadena para el número de teléfono, y un entero sin signo para llevar la edad de la persona.
7. Implemente una función **struct contacto crearcontacto(void)** que pida por teclado los datos pertinentes, rellene una estructura **contacto** y la devuelva.
8. Implemente una estructura para representar puntos en el plano, y una función **medio** que dados dos de estos puntos, calcule el punto medio.
9. Implemente una función **setin(int \*)** que toma un puntero a un entero, y reemplaza el entero apuntado por un 1 si el entero apuntado era diferente a 0, y 0 en caso contrario.
10. Implemente una función **void actualizaredad(struct contacto \*)** que dado un puntero a una estructura **contacto**, pida una nueva edad por teclado y actualice la estructura.
11. Implemente una función **int prom(struct contacto \*, int)** que reciba un puntero a estructuras **contacto**, y la cantidad de estructuras contiguas, y devuelva el promedio de la edad de estas estructuras. ¿Qué diferencia existe con tomar un arreglo de estructuras en lugar de un puntero?
12. Defina una estructura **agenda** que almacene un arreglo de estructuras **contacto**, y un entero para llevar la cantidad de estructuras almacenadas en el arreglo. Escriba un programa que permita: dar de alta un contacto, modificar la edad de un contacto, y ver los datos de los contactos cargados.
13. Escriba un programa que reserve un espacio de memoria de 100 bytes, y luego libérela dos veces. ¿Se produce algún error?
14. Implemente una función **int apply(int (\*)(int), int)** que toma un puntero a función, y un entero, y aplica la función al entero y retorna el valor dado.
15. Otra signatura válida para **main** es **int main(int argc, char \*argv[])**. El argumento **argc** cuenta la cantidad de argumentos pasados, y **argv** es un arreglo de cadenas que representan cada uno de los argumentos pasados al programa.  
Escriba un programa que imprime en pantalla la cantidad de argumentos que recibió, y el contenido de los mismos.