



Práctica 4

1. Punteros

1. Definir una función **swap** que dadas dos variables enteras intercambie el contenido de las mismas.
2. Definir versiones de las siguientes funciones sobre cadenas usando punteros:
 - i) **int strlen(char *c)**, que determina el largo de una cadena.
 - ii) **void strcpy(char *s, char *t)**, que copia la cadena **t** a **s**.

2. Estructuras

3. Definir una estructura **Punto** para representar puntos en el plano y una estructura **Rectangulo** que represente un rectángulo (suponer que un rectángulo está representado por el punto superior de más a la izquierda y el punto inferior de más a la derecha).
4. Definir una función **inRect**, que determine si un punto está dentro de un rectángulo.
5. Definir una estructura **Contacto** que tenga como campos: una cadena para el nombre de una persona y una cadena para el número de teléfono.
Implementar una función **crearContacto** que pida por teclado los datos de una persona y devuelva una estructura contacto con los mismos. Obtener memoria dinámicamente para guardar los datos de la persona.
6. Definir una estructura **Agenda** que almacene un arreglo de estructuras contacto, y un entero para llevar la cantidad de elementos del arreglo. Definir una función que permita agregar un contacto a la agenda y otra para ver los datos de todos los contactos.

3. Archivos

7. Definir una función con la siguiente signature: **void filecopy(FILE *ifp, FILE *ofp)**, que copie el contenido del archivo apuntado por **ifp** al archivo apuntado por **ofp**.
8. Definir una función que dado el nombre de un archivo pasado como argumento a la función, abra el archivo, cuente la cantidad de líneas, cree un vector de cadenas para almacenar las líneas del archivo y guarde en el vector todas las líneas del archivo.
9. Modificar la función definida en el apartado anterior para que la función retorne el vector con las líneas del archivo.