

## Temas dados para evaluar en el primer examen parcial teórico

### 1. Repaso de C:

- Programa básico: `include` (archivos cabecera), comentarios, `stdin/stdout`, función `main`, declaración de variables (donde?), tipos de datos básicos
- Operadores aritméticos, y de asignación
- Algoritmo, pseudo-código, diagrama de flujo
- Estructuras básicas: secuencia, selección (`if`, `if/else`, `switch`), repetición (`for`, `while`, `do/while`)
- Operadores condicionales y lógicos
- Operadores de incremento y decremento, asignación
- Funcionalización de programas. Prototipos y declaración de funciones. Parámetros, argumentos y valor de retorno. Pasaje de arreglos a funciones. Argumentos de `main`.
- 

### 2. Clases de almacenamiento, reglas de alcance, y calificadores de variables:

- Clases de almacenamiento, persistencia automática y estática.
- Palabras reservadas `auto`, `register`, `static`, y `extern`.
- Alcance de variables, ejemplos.
- Calificadores `volatile`, y `const`. Calificador `const` con punteros.

### 3. Caracteres y cadenas:

- Arreglos de caracteres/cadenas. Ejemplos.
- Biblioteca para el manejo de caracteres. Archivo de cabecera `ctype.h`.
- Funciones de conversión de cadenas. Archivo de cabecera `stdlib.h`.
- Funciones de manipulación y comparación de cadenas. Archivo `string.h`.
- Arreglo de punteros y arreglos de cadenas.

### 4. Punteros y estructuras:

- Repaso de punteros (filminas del Prof. Treviño).
- Estructura: definición, ejemplos, inicialización, operaciones válidas.
- `typedef` con estructuras.
- Arreglos y punteros de estructuras.
- Anidamiento de estructura.
- Pasaje a funciones.

### 5. Recursión:

1. Enfoque recursivo a la resolución de problemas. Conceptos y ejemplos de aplicación. Ventajas y desventajas.
2. Ejemplo de cálculo del factorial, y Fibonacci
3. Comparativa entre iteración y recursión.
4. Algunos comentarios sobre complejidad computacional.

### 6. Asignación dinámica de memoria:

- Organización en segmentos de la memoria de un programa en C (`text`, `data`, `stack`, y `heap`). Ejemplos.
- Funciones de la biblioteca estándar, `malloc`, `calloc`, `realloc`, `free`. Ejemplos y ejercicio.

7. Estructuras dinámicas de datos:
  - Estructuras auto-referenciadas.
  - Listas enlazadas: listas enlazadas, pilas, colas y árboles binarios.
  - Ejemplos detallados de estructuras de cada estructura de datos.
  
8. Uniones, manipuladores a nivel de bits y campos de bits (30/05):
  - Uniones, tipo de dato derivado, operaciones en uniones. Inicialización.
  - Ejemplo entero y float, float y unsigned char.
  - Operadores a nivel de bits, tabla de verdad. Ejemplo imprimir número en binario.
  - Operadores de desplazamiento a la izq. y der. Producto y división por 2.
  - Operadores a nivel de bits, y asignación.
  - Uso en manipulación de bytes, máscara, poner a cero, a uno, toggle de bits
  - Campos de bits.
  
9. GNU/Linux (13/06):
  - Introducción a GNU/Linux. Proyecto GNU. Licencias. Distribuciones.
  
10. Shell de Linux
  - Que es una shell? POSIX. Filosofía Linux (KISS). Comandos de usuario: whoami, groups, id (UID, GID). Autocompletar, TAB. Variables de entorno (PATH, HOME, SHELL). Manuales (1 y 3).
  - Sistema de archivos y permisos. Comandos: mkdir, rmdir, cd, ls, cp, mv, rm, touch, echo, which. Atributos de archivos (dispositivos: d, b, c), permisos, comando chmod.
  - Redirección de entrada/salida. Pipes.
  - Procesos. Comandos: ps, kill, pstree, top, htop.
  
11. El compilador de C del proyecto GNU (gcc, g++):
  - Herramienta de compilación gcc, ejemplos con uno y varios archivos fuentes.
  - Etapas de compilación: preprocesado, compilado, ensamblado, y enlazado. Ejemplos de construcción por etapas.
  - Macros y compilación condicional. Macros con valor, paso de valor por línea de comandos.
  - Bibliotecas: estática y dinámica. Uso de bibliotecas (archivos de cabecera y enlazado).
  - Flags de compilador -I y -L. Estándares del lenguaje C (flags).
  - Construcción de bibliotecas: estática y dinámica.

**Complementar el contenido con la filmas del Profesor Treviño**