# Primer parcial

Parte teórica

1. Defina que significa algoritmo, y que condiciones debe cumplir.

**Algoritmo es una secuencia de instrucciones de pasos finitos, los cuales se deben ser respetados y ejecutados paso a paso para poder resolver el problema planteado.**

1. Todo programa puede escribirse utilizando únicamente 3 estructuras básicas para el control de la programación. Mencione cada una ellas y dibuje su diagrama de flujo.

**Las 3 estructuras básicas para el control de la programación son: Estructura de control secuencial, selectiva y repetitiva. (Los diagramas de flujo están adjuntos en la carpeta de teoría).**

1. ¿En qué consiste el diseño **TOP-DOWN** y cuáles son sus ventajas?

**Consiste en la descomposición de un problema en subproblemas más simples. Sus ventajas son que mejoran mucho la legibilidad, se pueden reutilizar las funciones. Además las funciones se pueden escribir, compilar y depurar en forma independiente…En un programa de gran tamaño pueden trabajar distintos programadores. Se puede modificar (actualizar) una función sin afectar el resto del programa**

1. Indique cuáles son las distintas tecnologías en que se basa cada una de las generaciones de computadoras y las principales ventajas alcanzadas con la última respecto de la primera.

-

**No lo estudie.**

1. Explique el significado del modificador de ámbito ***static***. De ejemplos de su uso en lenguaje C cuando se utiliza en la declaración de una variable y una función.

**Mantiene el valor de una variable dentro de su ámbito (bloque, función o archivo). Cuando se aplica a una variable local por ejemplo, su valor se retiene entre sucesivas llamadas a la función (se inicializa automáticamente la primera vez con 0). Luego como modificador de funciones hace que la función sólo se puede acceder desde el archivo donde se declaró.**

1. Indique que frase se imprime al ejecutar el siguiente bloque de código y porqué.

**Este ejercicio lo tengo hecho y comentado en codeblocks y lo adjunto en la carpeta teoría.**

**uint8\_t** a = 0xFE;

**uint8\_t** par = a & 1;

**if** (par = 0)

printf("La variable es par");

**else**

printf("La variable es impar");

1. Indique que valor se imprime en la consola de salida luego de ejecutarse la sentencia printf(). Justifique este comportamiento.

**Este ejercicio lo tengo hecho y comentado en codeblocks y lo adjunto en la carpeta teoría.**

**uint8\_t** array[10];

**uint8\_t** value = 127;

**for** (**int** i = 0; i <= 10 ; i++)

{

array[i] = i;

}

printf("La variable valor es = %d \n", value);

1. Indique que se imprime en la consola de salida. Justifique su respuesta.

**Este ejercicio lo tengo hecho y comentado en codeblocks y lo adjunto en la carpeta teoría.**

#define PI 3.14159265358979323846

**const double** pi = 3.14159265358979323846;

(pi - PI) ? printf("pi es != PI \n") : printf("pi es = PI");

Parte práctica

1. Realice un programa en lenguaje C que intercambie los nibbles (grupo 4 bits) del byte 0xA1; por lo que la salida producida debe ser 26.
2. Se dispone de un algoritmo en pseudo-código realizado en PSeInt, cuyo listado se encuenta en el archivo *parcial\_01.psc*. Se pide realizar la versión en lenguaje C del mismo teniendo en cuenta las siguientes pautas:

* El programa debe estar correctamente modularizado, dividiendo el algoritmo presentado en la cantidad de funciones/archivos necesarios.
* La comunicación entre las diferentes funciones (paso de parámetros y valores de retorno) debe ser hecha de modo que la función receptora no pueda modificar valores críticos. Se evaluará la opción de pasar los mismos por valor o por referencia.
* Las funciones deben estar debidamente documentadas, indicando el significado y utilización de los parámetros recibidos y el valor de retorno.
* La compilación y linkeo finales deben concluir con *0 warnings*, *0 errors*.

## Aclaraciones

* **Para la aprobación del parcial se requiere que el programa entregado funcione según las pautas indicadas, y al menos el 50% de los ítems teóricos deben estar bien contestados.**
* Lea atentamente los enunciados de cada ejercicio. Durante la primera h se contestarán las dudas que surjan de los enunciados. Recuerde que primero deberá saber QUE hacer y luego pensar COMO hacerlo.
* Deberá entregar junto al proyecto de CodeBlocks este mismo documento, con las respuestas a las preguntas teóricas.
* Recuerde guardar el proyecto de CodeBlocks en la unidad D: y que el path no contenga espacios ni caracteres especiales como la ñ o las vocales acentuadas.

**Exitos! =)**