

## Trabajo Práctico Nivel 2

**Modularidad conceptos; Módulo concepto; Módulo clasificación en C; Funciones de biblioteca de C más utilizadas; Funciones definidas por el programador; Función: declaración, prototipo; Función: ámbito (Local, Global; Función: comunicación parámetros (Formales, Actuales, Referencia, Valor).**

**Fecha de Inicio 20 / 09 / 2021**

**Fecha de Fin 24 / 09 / 2021**

**Ejercicio 1:** Crear un programa que dados N números enteros, calcule la cantidad de dígitos de cada número. Si la cantidad de dígitos resulta ser un número primo entonces crear un nuevo número calculando la potencia del número ingresado, elevado al dígito más significativo. Si la cantidad de dígitos no es primo, crear un nuevo número calculando la raíz cuadrada del número. Mostrar cada número generado indicando el tipo de transformación.

**Nota:** la función de potencia y de raíz cuadrada se encuentran en la librería <math.h>

**Ejercicio 2:** Utilice la librería <stdlib.h> y <time.h> para crear un programa que simule la espera de N clientes en una cola para hacer un trámite en el banco. Para cada uno de los N clientes que llegan al banco, se asigna el tiempo de espera de forma aleatoria, siendo un número natural en el rango [1,30]. El programa debe calcular:

- Promedio total de espera
- El tiempo de la menor espera
- El tiempo de la mayor espera

**Ejercicio 3:** Dado un número natural determine si es un número deficiente, abundante o perfecto. **Nota:** Un número natural X se dice que es deficiente si la suma de sus divisores (sin contarse a sí mismo) es menor a X. Si la suma de sus divisores es mayor a X se dice que el número es abundante y si es igual a X se dice que es perfecto.

**Ejercicio 4:** Realizar un programa que genere una lista de N números aleatorios en el rango [A, B] (con A y B naturales) y muestre aquellos números aleatorios que cumplan con no ser deficientes.

**Ejercicio 5:** Realizar un programa que permita el ingreso de un número natural, y posteriormente calcule su equivalente a base 2, si la cantidad de divisores del número es primo y calcule su equivalente a base 9 si la cantidad de divisores no es primo.

**Ejercicio 6: Mejoramos el ejercicio 3 del TP de nivel 1**

- Generar un módulo para validar el tipo de vehículo. Para ello vamos ingresar el tipo de vehículo en una variable de tipo char donde solo se puede ingresar la letra mayúscula M para motocicletas, A para automóvil y C para camioneta. Este módulo solo va a retornar alguna de esas letras. Adaptar el programa para que utilice dicho módulo.
- Generar un módulo que calcule el importe a pagar por un vehículo y luego adecue el programa para que utilice dicho modulo.

**Ejercicio 7:** Dada N fechas ingresadas por el usuario, para cada una de ellas genere una fecha al azar mayor a esta y luego indique la cantidad de días transcurridos entre la fecha ingresada y la fecha generada al azar. También mostrar la cantidad promedio de días transcurridos.

**Ejercicio 8:** Realizar un programa que simule la evolución de un set de un partido de tenis. Recuerde que un set lo gana el jugador que primero llega a seis games (no se tendrá en cuenta el tie-break). Para simular el usuario se ingresará un número entero con los dígitos 1 y 2, que indican cual jugador ha ganado cada punto de un game (el número representa los datos del game). Para cada dígito del número se indicará el nuevo marcador del game. Se supondrá que el game lo gana el jugador que consigue cuatro puntos.

Considerar que:

- El primer punto que consigue el jugador es quince Cuando el jugador consigue el segundo punto es treinta.
- Cuando el jugador consigue el tercer punto es cuarenta.
- Cuando el jugador consigue el cuarto punto, gana el juego. Por ejemplo: si S=112211, la salida sería "quince-nada", "treinta-nada", "treinta-quince", "iguales treinta", "juego para el jugador 1".

Si S= 1221211 sería "quince-nada", "iguales quince", "quince-treinta", "iguales-treinta", "cuarenta, treinta", "iguales cuarenta", "juego para el jugador 1"