

TRABAJO PRÁCTICO N° 1 - Parte 2



Objetivos:

- Repasar el funcionamiento de la estructura Switch
- Aplicar conocimientos referidos a la estructura de la información
- Comprender el concepto de puntero, dato simple y dato estructurado
- Definir tipos de datos

ES OBLIGATORIO UTILIZAR NOMBRES DE VARIABLES SIGNIFICATIVOS

Parte 2 - Tema: Estructuras de Programación - Punteros - Datos Estructurados.

Problemas resueltos

Ej. 1 - Ingresar dos variables tipo estructura con los datos referentes a una fecha (día, mes, año). Luego presentar por pantalla el contenido de las variables.

```
#include <stdio.h>
```

```
struct fecha {
    int dia;
    int mes;
    int anio;
};

int main (int argc, char *argv[])
{
    struct fecha unaFecha, otraFecha={2, 10, 2008};
    unaFecha.dia= 10;
    unaFecha.mes = 12;
    printf("\n Ingresar anio:");
    scanf ("%d", &unaFecha.anio);

    printf("\n La primer fecha es: \n ");
    printf("\n Día: %d ", unaFecha.dia);
    printf("\n Mes: %d ", unaFecha.mes);
    printf("\n Anio: %d\n ", unaFecha.anio);
    printf("\n La otra fecha es: %d/%d/%d\n ", otraFecha.dia, otraFecha.mes,
    otraFecha.anio);
    return 0;
}
```

Ej. 2 - Definir una estructura en la que se represente la inicial del apellido y del nombre (1 carácter para cada uno) y la edad de una persona. Solicitar el ingreso de las iniciales y la edad por teclado. Presentar por pantalla las iniciales en mayúscula y la edad.

INVESTIGAR: cómo transformar un carácter de minúscula a mayúscula y viceversa. ¿Para qué sirve ctype.h?

```
#include <stdio.h>
#include <ctype.h>

struct persona {
    char nombre;
    char apellido;
    int edad;
};

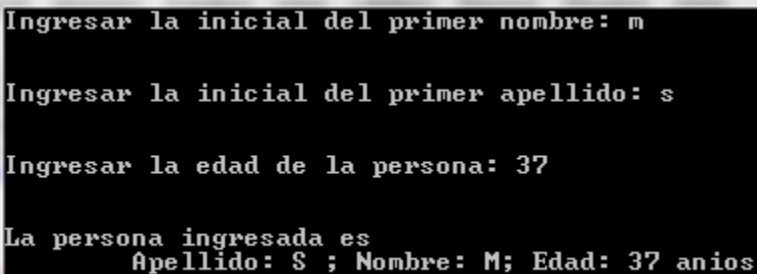
int main(int argc, char *argv[]){
    struct persona unaPersona;
    char c;
    system("clear");
    printf("Ingresar la inicial del primer nombre: ");
    scanf("%c", &unaPersona.nombre);
    fgetc(stdin);
    printf("\n\nIngresar la inicial del primer apellido: ");
    scanf("%c", &unaPersona.apellido);
    do{
        printf("\n\nIngresar la edad de la persona: ");
        scanf("%d", &unaPersona.edad);
    }while(unaPersona.edad<=0);

    unaPersona.nombre= toupper(unaPersona.nombre);
    c=toupper(unaPersona.apellido);

    printf("\n\nLa persona ingresada es\n\tApellido: %c ; Nombre: %c; Edad: %d
años" ,c, unaPersona.nombre,unaPersona.edad);

    return 0;
}
```

La salida por pantalla resulta:



```
Ingresar la inicial del primer nombre: m

Ingresar la inicial del primer apellido: s

Ingresar la edad de la persona: 37

La persona ingresada es
    Apellido: S ; Nombre: M; Edad: 37 años
```

Problemas para resolver

Codificar los programas que permitan resolver los siguientes ejercicios

1. Implementar un programa que solicite al usuario un número del 1 al 7, y luego muestre por pantalla a qué día de la semana corresponde, si no corresponde a ninguno mostrar un mensaje. Considerar el lunes el día 1 de la semana. **NO utilizar switch.** Por ejemplo, si se ingresa el valor 1, mostrar “El día ingresado es Lunes”, si se ingresa 15 mostrar “El valor ingresado no corresponde a un día de la semana”
2. Implementar un programa que solicite al usuario un número del 1 al 7, y luego muestre por pantalla a qué día de la semana corresponde, si no corresponde a ninguno mostrar un mensaje. Considerar el lunes el día 1 de la semana. **Utilizar switch.** Por ejemplo, si se ingresa el valor 1, mostrar “El día ingresado es Lunes”, si se ingresa 15 mostrar “El valor ingresado no corresponde a un día de la semana”
3. Implementar el programa del ejercicio anterior considerando que el usuario puede introducir las opciones del 1 al 8, en donde 1- Lunes, 2- Martes, 3 -Miércoles, 4- Jueves, 5- Viernes, 6-Sábado, 7-Domingo, 8-Salir. Ud debe mostrarlas en un menú y presentar por pantalla el mensaje correspondiente, solo debe salir del programa si el usuario presiona 8. **Sugerencia combinar do while y switch.**
4. Implementar un programa que solicite al usuario el ingreso de un signo del zodiaco de una lista de opciones y le muestre por pantalla un mensaje que indique si es de agua, aire, tierra, fuego o si ha seleccionado un valor inválido. **Sugerencia: usar switch.** Por ejemplo, si se ingresa el valor 1, mostrar “El signo de Aries es de Fuego”, si se ingresa 15 mostrar “El valor ingresado no corresponde a una opción válida”.

Signo	Categoría		
1. Aries	Fuego	7. Libra	Aire
2. Tauro	Tierra	8. Escorpio	Agua
3. Géminis	Aire	9. Sagitario	Fuego
4. Cáncer	Agua	10. Capricornio	Tierra
5. Leo	Fuego	11. Acuario	Aire
6. Virgo	Tierra	12. Piscis	Agua

5. Implementar el programa anterior como un menú de opciones, en donde el valor 13 es Salir. Por ejemplo, si se ingresa el valor 1, mostrar “El signo de Aries es de Fuego” y mostrar nuevamente el menú, si se ingresa 15 mostrar “El valor ingresado no corresponde a una opción válida”, si se ingresa 13 se debe salir del programa. **Sugerencia combinar do while y switch.**
6. Implementar un programa que ayude a una empresa a calcular el nuevo sueldo de los trabajadores, al incrementarse el mismo de la siguiente manera:
Si el empleado tiene un contrato con un sueldo:
 - a. De 0 a \$35 000 se incremente un 20%
 - b. De \$35 000 a \$65 000 se incremente un 10%

- c. De \$65 000 a \$950 000 se incremente un 5%
- d. Más de \$95 000 se incremente un 3%

Se pide: Calcular y mostrar por pantalla el nuevo sueldo de acuerdo al contrato del empleado.

*Resolver los siguientes problemas utilizando variables tipo **puntero***

7. Ingresar un número entero, luego asignar la dirección de memoria de dicha variable a una variable puntero. Presentar por pantalla la información que contiene la variable puntero (dirección e indirección) y la variable entera almacenada.
8. Ingresar un número entero, luego utilizando una variable tipo puntero, presentar por pantalla el valor ingresado, el domicilio de memoria en donde se aloja dicho valor y el domicilio de memoria del puntero utilizado.
9. Ingresar un número entero, luego presentar por pantalla el valor duplicado de dicho Número.
10. Cargar en una variable index un valor constante, declarar dos punteros. Asignarle al primer puntero la dirección de memoria de la variable index, luego copiar el contenido del primer puntero al segundo puntero declarado. Mostrar por pantalla el contenido de cada una de las variables.
11. Realizar un programa que solicite por teclado, dos números enteros. Luego realizar las operaciones de suma, resta, división real, división entera, resto de la división entera y multiplicación entre ambos valores ingresados, recordar que en la división, el denominador no debe ser igual a cero. Presentar por pantalla los resultados.
12. Se ingresan N valores numéricos. Se desea saber cuántos son positivos, cuántos negativos y cuantos iguales a cero.
13. Se ingresan N valores numéricos. Determinar y presentar por pantalla el mayor y el menor de los valores ingresados.

*Resolver los siguientes problemas utilizando variables tipo **estructura***

Importante: *para la definición del tipo de dato puede usar struct o hacerlo empleando typedef struct*

14. Ingresar un número real base y un número entero positivo exponente presentar por pantalla el valor de base elevado al valor de exponente. Utilizar una estructura para representar el par de valores base y exponente.
15. Implementar un programa en el que se soliciten el radio de la base y la altura de un cilindro circular y luego se presenten por pantalla. Usar una estructura como la siguiente:

```
struct cilindro{
    float radio;
    float altura;
};
```

16. Implementar un programa en el que se ingresan los datos de N personas. Calcular y presentar por pantalla el promedio de edades de las personas ingresadas. Utilizar la estructura en la que se represente la inicial del apellido y del nombre (1 caracter para cada uno) y la edad de una persona. (Ej. 2)

```
struct persona {
    char nombre;
    char apellido;
    int edad;
};
```

17. Definir una estructura cuyos campos permitan guardar la cantidad de caracteres que sean vocales y la cantidad que sean consonantes. Luego debe ingresar N caracteres, y contabilizar en la estructura definida los caracteres de acuerdo a su tipo (vocales o consonantes). *Nota: Para resolver es importante Investigar! ¿Qué librería se requiere para el uso de funciones que permitan evaluar caracteres? ¿Para qué usamos el fgetc(stdin)? ¿Qué otra instrucción se podrá utilizar?*

18. Codificar un programa para: ingresar dos números N y D enteros positivos. Debe realizar el control del ingreso de datos, luego calcular el valor del cociente y resto entre N y D utilizando el método de restas sucesivas. Para almacenar y mostrar por pantalla el resultado utilizar la siguiente estructura:

```
struct dividir{
    float cociente;
    int resto;
};
```

19. Ingresar un conjunto no establecido de números, la carga debe terminar cuando se ingrese un cero; luego presentar por pantalla el menor y el mayor valor de los números ingresados, indicando el orden en que ingresaron. Utilizar una estructura como la siguiente para almacenar los resultados:

```
struct num{
    int valor;
    int posicion;
};
```

20. Implementar un programa que solicite la fecha actual (almacenando en una variable estructura) Luego solicite y almacene en una estructura las iniciales y la fecha de nacimiento de una persona (ejemplo 20/6/1999). Para finalizar presente por pantalla la edad en años de la persona, por ejemplo: *La edad de S, M es xx años*. Es importante controlar el valor de los datos de entrada, por ejemplo los meses deben ir entre 1 y 12.

21. Codificar un programa para ingresar la altura, el ancho y largo de una caja y los almacene en una estructura. Luego calcular y presentar por pantalla el volumen de la caja.

22. Ingresar N números (realizar el control de que N sea positivo), contabilizar cuántos números pares, impares y primos se encontraron, luego mostrar los resultados por pantalla. Utilizar una estructura como la siguiente:

```
struct categorias{
    int pares;
    int impares;
    int primos;
};
```

23. En una fábrica se registra la información del mantenimiento de las N máquinas que posee la fábrica en una estructura como la siguiente:

```
struct maquina{
    int numeroMaquina;
    char sector;
```

```
    int ultimoAñoMantenimiento;
};
```

Se pide calcular y mostrar el porcentaje de máquinas cuyo último año de mantenimiento fue en el 2020.

24. Implementar un programa para un hospital que registre el ingreso de pacientes a ser atendidos por COVID-19, el ingreso termina cuando se presione la tecla n. Utilizar una estructura como la siguiente:

```
struct paciente {
    int codigoPaciente;
    char nombre;    (guardar un carácter)
    char apellido;  (guardar un carácter)
    char resultadoTest; (P: positivo, N: negativo)
};
```

Se pide calcular y mostrar por pantalla el porcentaje con resultado del test positivo.

25. Implementar un programa para un hospital que registre el ingreso de pacientes a ser atendidos por COVID-19, el ingreso termina cuando se presione la tecla n o N. Utilizar una estructura como la siguiente:

```
struct paciente {
    int codigoPaciente;
    char nombre;    (guardar un carácter)
    char apellido;  (guardar un carácter)
    fecha fechaNacimiento (estructura con día, mes, año)
    char resultadoTest; (P: positivo, N: negativo)
};
```

Se pide contar y mostrar por pantalla la cantidad de pacientes mayores a 60 años con resultado del test positivo.