



Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño



Ingeniero en Software y Tecnologías Emergentes.

Programación estructurada

LIBRERÍAS, CADENAS Y FUNCIONES

Actividad 9 y 3/4

Danna Guadalupe Sandez Islas

373080

```
#include "puente.h"

typedef struct _nombre
{
    char ap_paterno[50];
    char ap_materno[50];
    char nombre[50];
} Tnombre;

typedef struct _fecha
{
    int mes;
    int dia;
    int anio;
} Tfecha;

typedef struct _datos
{
    Tnombre nom;
    Tfecha nacimiento;
    int estado;
    int sexo;
} Tdatos;

/* Prototipo de funciones*/
Tdatos data_curp(void);
void print_curp(Tdatos persona);
/* Funciones auxiliares*/
```

Main:

```
int main()
{
    int op;
    Tdatos persona;
    do
    {
        system("CLS");
        printf("--- Generador de CURP ---\n");
        printf("Quieres generar tu CURP?\n");
        printf("1-Si\n2-No\n");
        op = validar("Ingresa una opcion numerica: ", 1, 2);
        system("CLS");
        if (op == 1)
        {
            persona = data_curp();
            printf("\nTu CURP es: ");
            print_curp(persona);
            printf("\n");
            system("PAUSE");
        }
    } while (op != 2);
    printf("Gracias por usar el programa. Hasta luego!");

    return 0;
}
```

Función para obtener datos:

```
Tdatos data_curp(void)
{
    Tdatos humano;

    int op = 0;
    int op2 = 0;
    // nombre
    op = validar("Tienes apellido paterno?\n1-Si\n2-No\nIngresa una opcion: ", 1, 2);
    system("CLS");
    if (op == 1)
    {
        cadena("Apellido paterno: ", humano.nom.ap_paterno);
    }
    else
    {
        noAP(humano.nom.ap_paterno);
    }

    system("CLS");

    op2 = validar("Tienes apellido materno?\n1-Si\n2-No\nIngresa una opcion: ", 1, 2);
    system("CLS");
    if (op2 == 1)
    {
        cadena("Apellido materno: ", humano.nom.ap_materno);
    }
    else
    {
        noAP(humano.nom.ap_materno);
    }

    system("CLS");

    cadena("Nombre: ", humano.nom.nombre);
```

```
    // fecha de nacimiento
    humano.nacimiento.anio = validar("Año de nacimiento AAAA: ", 1893, 2023);
    system("CLS");
    if (humano.nacimiento.anio == 2023)
    {
        humano.nacimiento.anio = validar("Mes de nacimiento MM: ", 1, 10);
    }
    else
    {
        humano.nacimiento.mes = validar("Mes de nacimiento MM: ", 1, 12);
    }

    humano.nacimiento.dia = dia_n(humano.nacimiento.mes, humano.nacimiento.anio);
    system("CLS");
    // sexo
    humano.sexo = validar("      ----SEXO----\n1-Mujer\n2-Hombre\nIngresa una opcion NUMERICA: ", 1, 2);
    system("CLS");
    // Estado
    printf("      ----ESTADO----\n");
    humano.estado = estados();
    system("CLS");

    return humano;
}
```

Función para imprimir CURP:

```
void print_curp(Tdatos persona)
{
    int apComp1 = 0;
    int apComp2 = 0;
    int nomComp = 0;
    char clave[4];
    // hace mayusculas los nombres
    mayus(persona.nom.ap_paterno);
    mayus(persona.nom.nombre);
    mayus(persona.nom.ap_materno);

    // saber si tiene apellido compuesto
    if (strlen(persona.nom.ap_paterno) != 0)
    {
        if (nomCompuesto(persona.nom.ap_paterno) == 1) // tiene ap compuesto
        {
            apComp1 = 1;
            carctEspecial(persona.nom.ap_paterno, clave);
            enie(persona.nom.ap_paterno);
            dieresis(persona.nom.ap_paterno);
            apeCompuesto(persona.nom.ap_paterno, clave);
        }
        else // no tiene apellido compuesto
        {
            // carctEspecial(persona.nom.ap_paterno, clave);
            ape(persona.nom.ap_paterno, clave);
        }
    }
    else
    {
        clave[0] = 'X';
        clave[1] = 'X';
    }
}
```

```

// imprimir letra inicial del segundo apellido
if (strlen(persona.nom.ap_materno) != 0) // checa si tiene apellido materno
{
    if (nomCompuesto(persona.nom.ap_materno) == 1) // tiene ap compuesto
    {
        apComp2 = 1;
        carctEspecial(persona.nom.ap_materno, clave);
        enie(persona.nom.ap_materno);
        dieresis(persona.nom.ap_materno);
        apeCompuestoMa(persona.nom.ap_materno, clave);
    }
    else // no tiene ap compuesto
    {
        apeMa(persona.nom.ap_materno, clave);
    }
}
else
{
    clave[2] = 'X';
}

if (nomCompuesto(persona.nom.nombre) == 1) // tiene nombre compuesto
{
    nomComp = 1;
    enie(persona.nom.nombre);
    dieresis(persona.nom.nombre);
    mariajose(persona.nom.nombre, clave);
}
else
{
    carctEspecial(persona.nom.nombre, clave);
    enie(persona.nom.nombre);
    dieresis(persona.nom.nombre);
    clave[3] = persona.nom.nombre[0];
}

```

```
// verificar si es una clave antisonante
if (antisonante(clave) == 1)
{
    for (int i = 0; i < 4; i++)
    {
        printf("%c", clave[i]);
    }
}
else
{
    for (int i = 0; i < 4; i++)
    {
        printf("%c", clave[i]);
    }
}

// imprimir fecha de nacimiento
printf("%02d%02d%02d", persona.nacimiento.anio % 100, persona.nacimiento.mes, persona.nacimiento.dia);

// imprimir sexo
if (persona.sexo == 1)
{
    printf("%s", "M");
}
else
{
    printf("%s", "H");
}

// imprimir estado
if (persona.estado < 32)
{
    print_est(persona.estado);
}
else
{
    printf("%s", "NE");
}
```

```

// imprimir consonantes del apellido paterno

if (strlen(persona.nom.ap_paterno) != 0)
{
    if (apComp1 == 1) // primer apellido
    {
        // el apellido es compuesto
        consCompuesto(persona.nom.ap_paterno);
    }
    else
    {
        consonante(persona.nom.ap_paterno);
    }
}
else
{
    printf("%c", 'X');
}
// imprimir consonante de apellido materno
if (strlen(persona.nom.ap_materno) != 0)
{
    if (apComp2 == 1) // materno
    {
        consCompuesto(persona.nom.ap_materno);
    }
    else
    {
        if (consonante(persona.nom.ap_materno) == 0)
        {
            consNombre(persona.nom.ap_materno);
        }
    }
}
else // no tiene apellido materno
{
    printf("%c", 'X');
}

```



```

// imprimir primer consonante del nombre
if (nomComp == 1) // nombre
{
    consCompuesto(persona.nom.nombre);
}
else
{
    if (consonante(persona.nom.nombre) == 0)
    {
        consNombre(persona.nom.nombre);
    }
}

// imprimir diferenciador de nacimiento
if (persona.nacimiento.anio <= 1999)
{
    printf("%d", 0);
}
else
{
    printf("%s", "A");
}

// imprimir digito verificador
printf("%d", numAleatorio(0, 9));
}

```

DATOS PARA PROBAR FUNCIONAMIENTO DE CURP

FRAIRE

LOPEZ

JOSE

OMAR

1997

31

ENERO

HOMBRE

GUERRERO

CURP: FXLO970131HGRRPM02

Tu CURP es: FXLO970131HGRRPM07
 Presione una tecla para continuar . . . █

CHAN

CWEN

2002

MARZO


28

HOMBRE

NACIDO EN EL EXTRANJERO

CURP: CAXC020328HNEHXWA2

Tu CURP es: CAXC020328HNEHXWA0

Presione una tecla para continuar . . . 

DE LA LOMA

XIU

JOSE

ANDRES

2007

DICIEMBRE


28

HOMBRE

YUCATAN

CURP: LOXA071228HYNMXNA3

Tu CURP es: LOXA071228HYNMXNA9

Presione una tecla para continuar . . . 

Funciones para validación se encuentran en el archivo: puente.h