



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 11:

FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA

ESTRUCTURAS Y LIBRERIAS (p2)

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 3 de Noviembre del 2023

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

```
#define MESES 12
#define ESTADOS 33
```

```
typedef struct _nombre{
    char nombres[50];
    char apellido1[50];
    char apellido2[50];
} Tnombre;
```

```
typedef struct _fecha{
    int anio;
    int mes;
    int dia;
    char anio1[5];
    char mes1[3];
    char dia1[3];
} Tfecha;
```

```
typedef struct _datos{
    int status;
    int matricula;
    Tnombre _nombre;
    int edad;
    Tfecha _fecha;
    int sex;
    char sexo[2];
    char estado[33];
```

```
    char estado[33];
    char estado_abreviado[33];
} Tdatos;
```

```
Tnombre generar_nombre_random(char nombre[]);
Tfecha generar_fecha_random(void);
Tdatos generar_datos_random(void);
```

```
int rango(int ri, int rf);
int matricula_random();
int bisiestro_random(int anio);
int ValidarCadena(char mensj[], int ri, int rf);
int ValidarCadenaTexto(const char cadena[]);
int ValidarGenero(const char genero[2]);
int ValidarRango(int num, int ri, int rf);
int ValidarNombre(char cadena[]);
int Mayusculas(char cadena[]);
int validar_fecha(int anio, int mes, int dia);
int contador(char array[]);
```

```
void meses_delanio(char meses[][30]);
void nombre_femenino1(char cadena[]);
void nombre_femenino2(char cadena[]);
void nombre_masculino1(char cadena[]);
void nombre_masculino2(char cadena[]);
void apellidos(char cadena[]);
void segundo_nombre_masculino(char cadena[]);
void segundo_nombre_femenino(char cadena[]);
void estados(char estado_abreviado[], int opcion, char estado[]);
void estados_republica(char estados[][30]);
void estados_republica_abreviados(char estados_abreviados[][3]);
```

```

int rango(int ri, int rf)
{
    return ri + rand() % (rf - ri + 1);
}

int matricula_random()
{
    int ri = 300000;
    int rf = 399999;
    return ri + rand() % (rf - ri + 1);
}

int bisiestro_random(int anio)
{
    return ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0));
}

int ValidarCadena(char mensj[], int ri, int rf)
{
    int num;
    char cadena[200];
    do
    {
        printf("%s", mensj);
        fflush(stdin);
        gets(cadena);
        num = atoi(cadena);
    } while (num < ri || num > rf);
    return num;
}

```

```

}

int ValidarGenero(const char genero[2])
{
    return (strcmp(genero, "H") == 0 || strcmp(genero, "M") == 0);
}

int ValidarRango(int num, int ri, int rf)
{
    return (num >= ri && num <= rf);
}

int ValidarNombre(char cadena[])
{
    int i = 0;

    if (cadena[0] == ' ')
    {
        return -1;
    }

    if (cadena[0] == '\\0')
    {
        return -1;
    }

    for (i = 0; cadena[i] != '\\0'; i++)
    {
        if (cadena[i] == -92 || cadena[i] == -91 || cadena[i] == '/' || cadena[i] == '-')
        {
            cadena[i] = 'X';
        }
    }
}

```

```

        {
            if ( cadena [i] <= 'z' )
            {
                cadena[i] = cadena[i] - ( 'a' - 'A' );
            }
        }
        i++;
    }
    return 0;
}

int validar_fecha(int anio, int mes, int dia)
{
    if (mes == 2 && dia == 29)
    {
        if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0))
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }

    if ((mes == 4 || mes == 6 || mes == 9 || mes == 11) && dia > 30)
    {
        return 0;
    }

    if ((mes == 1 || mes == 3 || mes == 5 || mes == 7 || mes == 8 || mes == 10 || mes == 12) && dia > 31)
    {

```

```

printf("Matricula: %d\n", _datos.matricula);
printf("Nombres: %s\n", _datos._nombre.nombres);
printf("Primer apellido: %s\n", _datos._nombre.apellido1);
printf("Segundo apellido: %s\n", _datos._nombre.apellido2);
printf("Dia: %d\n", _datos._fecha.dia);
printf("Mes: %d\n", _datos._fecha.mes);
printf("Año: %d\n", _datos._fecha.ano);
printf("Edad: %d\n", _datos.edad);
printf("Sexo: %s\n", _datos.sexo);
printf("Estado: %s\n", _datos.estado);

return _datos;
}

Tdatos generar_datos_manual(Tdatos _datos)
{
    int op;
    char PriNom[100];
    char SegNom[100];
    char PriApe[100];
    char SegApe[100];

    int ap;
    char anio[5];
    char mes[3];
    char dia[3];

    int up;
    int seleccion;
    char input[20];
    char meses[12][30];

```

```

_datos.matricula = ValidarCadena("Ingresa la matricula (entre 300000 y 399999): ", 300000, 399999);

do {
    printf("Ingresa tu primer nombre: \n");
    fflush(stdin);
    gets(PriNom);
    Mayusculas(PriNom);
    op = ValidarCadenaTexto(PriNom);
    if (op != 1)
    {
        printf("Nombre no valido.\n");
    }
} while (op != 1);
strcpy(_datos._nombre.nombres, PriNom);
op = ValidarCadena("Tienes un segundo nombre? (SI = 0, NO = 1): \n", 0, 1);
if (op == 0)
{
    printf("Ingresa tu segundo nombre: \n");
    fflush(stdin);
    gets(SegNom);
    Mayusculas(SegNom);
    op = ValidarCadenaTexto(SegNom);
    if (op != 1)
    {
        printf("Nombre no valido.\n");
    }
    strcat(_datos._nombre.nombres, " ");
    strcat(_datos._nombre.nombres, SegNom);
}

do {

```

```

do {
    ap = ValidarCadena("Ingresa tu anio de nacimiento (4 digitos): \n", 1893, 2023);
    if (ap < 1893)
    {
        printf("El año debe ser mayor o igual a 1893.\n");
    }
    else
    {
        if (ap > 2023)
        {
            printf("El año debe ser menor o igual a 2023.\n");
        }
    }
} while (ap < 1893 || ap > 2023);
sprintf(_datos._fecha.anio1, "%d", ap);

printf("Meses del anio:\n");
for (int i = 0; i < MESES; i++)
{
    printf("%d. %s\n", i + 1, meses[i]);
}

do {
    printf("Ingresa tu mes de nacimiento (numero): ");
    fflush(stdin);
    gets(input);
    seleccion = atoi(input);
    up = ValidarRango(seleccion, 1, MESES);
    if (up != 1)
    {
        printf("Mes no valido. Debe estar entre 1 y %d.\n", MESES);
    }
} while (up != 1);

```

```

void meses_delanio(char meses[][30])
{
    strcpy(meses[0], "Enero");
    strcpy(meses[1], "Febrero");
    strcpy(meses[2], "Marzo");
    strcpy(meses[3], "Abril");
    strcpy(meses[4], "Mayo");
    strcpy(meses[5], "Junio");
    strcpy(meses[6], "Julio");
    strcpy(meses[7], "Agosto");
    strcpy(meses[8], "Septiembre");
    strcpy(meses[9], "Octubre");
    strcpy(meses[10], "Noviembre");
    strcpy(meses[11], "Diciembre");
}

void nombre_femenino1(char cadena[])
{
    char PriNomFem[10][10] = {"MARIA", "ANA", "LAURA", "ISABEL", "CARMEN", "MARTHA", "PATRICIA", "TERESA", "ELENA", "ROSA"};
    strcpy(cadena, PriNomFem[rand()%10]);
}

void nombre_femenino2(char cadena[])
{
    char SegNomFem[10][15] = {"BEATRIZ", "DANNA", "SARA", "SOFIA", "LUCIA", "ALONDRA", "GABRIELA", "CARLA", "ANA", "PAULA"};
    strcpy(cadena, SegNomFem[rand()%10]);
}

void nombre_masculino1(char cadena[])
{
    char PriNomMas[10][10] = {"JUAN", "MANUEL", "JESUS", "CARLOS", "ALEJANDRO", "BERNABO", "SAUL", "JAVIER", "MIGUEL", "ALBERTO"};

```

```

    strcpy(cadena, PriNomMas[rand()%10]);
}

char estado_pequeno[33][3] = {
    "AS", "BC", "BS", "CC", "CL", "CM", "CS", "CH", "DF", "DG", "GT", "GR",
    "HG", "JC", "MC", "MN", "MS", "NI", "NL", "OC", "PL", "QT", "QR", "SP",
    "SL", "SR", "TC", "TS", "TL", "VZ", "YN", "ZS", "NE"};

char estado_grande[33][25] = {
    "AGUASCALIENTES", "BAJA CALIFORNIA", "BAJA CALIFORNIA SUR", "CAMPECHE",
    "CHIAPAS", "CHIHUAHUA", "COAHUILA", "COLIMA", "CIUDAD DE MEXICO",
    "DURANGO", "GUANAJUATO", "GUERRERO", "HIDALGO",
    "JALISCO", "MEXICO", "MICHOACAN", "MORELOS", "NAYARIT",
    "NUEVO LEON", "OAXACA", "PUEBLA", "QUERETARO",
    "QUINTANA ROO", "SAN LUIS POTOSI", "SINALOA", "SONORA",
    "TABASCO", "TAMAULIPAS", "TLAXCALA", "VERACRUZ",
    "YUCATAN", "ZACATECAS", "EXTRANJERO"};

strcpy(estado_abreviado, estado_chiquito[opcion]);
strcpy(estado, estado_grande[opcion]);
}

void estados_republica(char estados[][30])
{
    strcpy(estados[0], "Aguascalientes");
    strcpy(estados[1], "Baja California");
    strcpy(estados[2], "Baja California Sur");
    strcpy(estados[3], "Campeche");
    strcpy(estados[4], "Chiapas");
    strcpy(estados[5], "Chihuahua");
    strcpy(estados[6], "Coahuila");
    strcpy(estados[7], "Colima");
    strcpy(estados[8], "Ciudad de Mexico");
    strcpy(estados[9], "Durango");
    strcpy(estados[10], "Guanajuato");

```