



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Materia:

Programación Estructurada

Actividad:

11 - Anexos

Alumno:

Miguel Angel Ramirez Monjaraz

Matricula:

372205

Grupo:

Ingeniero en Computación - 432

Profesor:

Pedro Nunez Yepiz

Ensenada, Baja California a 31 de Octubre del 2023.

SALIDAS DE PANTALLA:

| MATRIUCLA: 331783
| NOMBRE: MARTA
| SEGUNDO NOMBRE: ISABEL
| APELLI PATERNO: SANCHEZ
| APELLI MATERNO: REYES
| FECHA DE NACIM: 04-02-1977
| EDAD: 46
| SEXO: MUJER
| LUGAR DE NACIM: HIDALGO
| CURP GENERADA : SARM770204MHGNYR4
|

| MATRIUCLA: 332270
| NOMBRE: PEDRO
| SEGUNDO NOMBRE: MIGUEL
| APELLI PATERNO: SANCHEZ
| APELLI MATERNO: VAZQUEZ
| FECHA DE NACIM: 24-09-1930
| EDAD: 93
| SEXO: HOMBRE
| LUGAR DE NACIM: CHIHUAHUA
| CURP GENERADA : SAVP300924HCHNZD8
|

| MATRIUCLA: 332745
| NOMBRE: ANA
| SEGUNDO NOMBRE: CARMEN
| APELLI PATERNO: SANCHEZ
| APELLI MATERNO: GUTIERREZ
| FECHA DE NACIM: 23-07-1978
| EDAD: 45
| SEXO: MUJER
| LUGAR DE NACIM: NUEVO LEON
| CURP GENERADA : SAGA780723MNLNTN4
|

```

//                               Ensenada, Baja California a 03 de Noviembre del 2023 //
// Nombre:                       Miguel Angel Ramirez Monjaraz //
// Matricula:                     372205 //
// Programa:                      Genera registros de personas con su curp correspondiente //
// Nombre del programa: MARM_ACT11_01 //
#include "curp.h"
#define N 2000

//DEFINIENDO ESTRUCTURAS//
typedef struct _alum
{
    char nombrel[30];
    char nombre2[30];
    char appater[30];
    char apmater[30];
}Nombre;

typedef struct _fnaci
{
    char anio[5] ;
    char mes[3] ;
    char dia[3];
}Datos;

typedef struct _registro
{
    int status;
    int matri;
    Nombre persona;
    Datos nacimiento;
    char estado[3];
    char estado_compl[30];
    char sexo[10];
    int edad;
    char curp[18];
}Registro;

//DEFINIENDO FUNCIONES//
Registro gen_rega_alea();
Registro eliminar_regis(Registro vect[], int *n);

void imprimir_registros(Registro vect[], int n);
void ordenar_regis(Registro vect[], int n);
void generarFechaAlea(char anio[],char mes[],char dia[]);
void buscar_regis(Registro vect[], int n);

void menuBusc(Registro vect[],int i,bool orden);
void letrero();
void menu();

int archivoTxt(Registro vect[],int i,int band);
int genMatri(Registro vect[],int i);
int buscarAlumSec(Registro vect[],int i,int matricula);
int randomDia(int mes, int bisiesto);
int buscarAlumBin(Registro vect[],int i,int matricula);
int esBisiesto(int year);
int calcularEdad(char anio1[],char mes1[],char dia1[]);

int main()
{
    menu();
}

//CUERPO DE LAS FUNCIONES

void letrero()
{
    printf("
                               MENU
                               \n");
    printf("1 - CARGAR (AUTOM 100 REGISTROS)\n");
    printf("2 - ELIMINAR\n");
    printf("3 - BUSCAR \n");
    printf("4 - ORDENAR\n5 - IMPRIMIR\n6 - ARCHIVO DE TEXTO\n0 - SALIR\n");
}

void menu()
{
    int txtband=1;
    int x = 0;
    int op;
    int i=0;
    int j = 100;
    bool orden=false;
    bool ordenado = true;
    Registro vect[N];
    do
    {
        letrero();
        op = vali_rango("Ingresa la accion a realizar: ",0,6);
        switch (op)
        {
            case 0:
                op = 0;
                break;
            case 1:
                if (j <= 1900)
                {
                    for (i; i < j ; i++)
                    {
                        int num = genMatri(vect,i);
                        vect[i] = gen_rega_alea();
                        vect[i].matri = num;
                    }
                    printf("Se han agregado 100 datos exitosamente!");
                }
                else
                {
                    printf("El registro se ha llenado completamente, intente realizar otra opcion, buen dia!\n");
                }
                j = j + 100;
                i = j - 101;
                break;
            case 2:
                int op;
                op= vali_rango("Deseas eliminar el registro?",1,2);
                if (op == 1)
                {
                    eliminar_regis(vect, &i);
                }
            }
        }
    } while (op != 0);
}

```

```

    }
    else
    {
        printf("Te has salvado");
    }
    break;
case 3:
    menuBusc(vect,i,orden);
    break;
case 4:
    if (ordenado)
    {
        ordenar_regis(vect, i);
        ordenado = false;
    }
    else
    {
        printf("Ya se encuentra ordenado");
    }
    break;
case 5:
    if (i>0)
    {
        imprimir_registros(vect, i);
    }
    else
    {
        printf("\nAun no hay personas registrados\n");
    }
    break;
case 6:
    if (i>0)
    {
        txtband=archivoTxt(vect,i,txtband);
        printf("Se genero un archivo de texto de los registros\n");
    }
    else
    {
        printf("\nAun no hay personas registrados\n");
    }
    break;
default:
    break;
}
} while (op != 0);
}

void imprimir_registros(Registro vect[], int n)
{
    char enter[2];
    for (int i = 0; i < n; i++)
    {
        printf("-----\n");
        printf("|                                \n");
        printf("| MATRIUCLA: %d                \n",vect[i].matri);
        printf("| NOMBRE: %s                   \n",vect[i].persona.nombre1);
        printf("| SEGUNDO NOMBRE: %s          \n",vect[i].persona.nombre2);
        printf("| APELLI PATERNO: %s          \n",vect[i].persona.apater);
        printf("| APELLI MATERNO: %s          \n",vect[i].persona.apmater);
        printf("| FECHA DE NACIM: %s-%s-%s    \n",vect[i].nacimiento.dia,vect[i].nacimiento.mes,vect[i].nacimiento.anio);
        printf("| EDAD: %d                    \n",vect[i].edad);
        printf("| SEXO: %s                    \n",vect[i].sexo);
        printf("| LUGAR DE NACIM: %s          \n",vect[i].estado_compl);
        printf("| CURP GENERADA : %s          \n",vect[i].curp);
        printf("|                                \n");
        printf("-----\n");
        printf("\n");
        if ((i + 1) % 40 == 0)
        {
            printf("-----\n");
            printf("----- Fin del grupo de 40 elementos ----- \n");
            printf("Presione cualquier tecla para continuar.....");
            fflush(stdin);
            gets(enter);
        }
    }
}

Registro gen_rega_alea()
{
    Registro regis;
    char nHombre[10] = {"Erick", "Juan", "Pedro", "Luis", "Carlos", "Miguel", "Alejandro", "JosÃ©", "Javier", "Antonio"};
    char nMujer[10] = {"Maria", "Ana", "Laura", "Sofia", "Paula", "Isabel", "Carmen", "Patricia", "Raquel", "Marta"};
    char apPater[10] = {"Gonzalez", "Lopez", "Rodriguez", "Martinez", "Perez", "Fernandez", "Gomez", "Sanchez", "Ramirez", "Torres"};
    char apMater[10] = {"Flores", "Morales", "Vazquez", "Jimenez", "Reyes", "Diaz", "Torres", "Gutierrez", "Ruiz", "Mendoza"};
    char a[5] = "\0";
    char m[3] = "\0";
    char d[3] = "\0";

    regis.status = 1;
    int sexo = rand() % 200;
    int dos_nom = rand() % 200;
    if (sexo%2 == 1)
    {
        if (dos_nom%2 == 1)
        {
            strcpy(regis.persona.nombre1, nHombre[rand() % 10]);
            strcpy(regis.sexo, "HOMBRE" );
        }
        else
        {
            strcpy(regis.persona.nombre1, nHombre[rand() % 10]);
            strcpy(regis.persona.nombre2, nHombre[rand() % 10]);
            strcpy(regis.sexo, "HOMBRE" );
        }
    }
    else
    {
        if (dos_nom%2 == 0)
        {
            strcpy(regis.persona.nombre1, nMujer[rand() % 10]);
            strcpy(regis.sexo, "MUJER" );
        }
        else
        {
            strcpy(regis.persona.nombre1, nMujer[rand() % 10]);
            strcpy(regis.persona.nombre2, nMujer[rand() % 10]);
            strcpy(regis.sexo, "MUJER" );
        }
    }
}

generarFechaAlea(a,m,d);

```

```

strcpy(regis.persona.appater, apPater[rand() % 10]);
strcpy(regis.persona.apmater, apMater[rand() % 10]);
strcpy(regis.nacimiento.anio, a );
strcpy(regis.nacimiento.mes, m);
strcpy(regis.nacimiento.dia, d );
regis.edad = calcularEdad(a,m,d);
genEst(regis.estado, regis.estado_compl);
generar_cup(regis.cup, regis.persona.nombre1, regis.persona.nombre2, regis.persona.appater, regis.persona.apmater, regis.nacimiento.dia, regis.nacimiento.mes, regis.nacimiento.anio, reg
return regis;
}

/////
Registro eliminar_regis(Registro vect[], int *n)
{
    int matri = vali_rango("Ingresa la matricula del registro que deseas borrar: ", 300000, 399999);
    int k = 0;
    Registro veck[N];

    for (int i = 0; i < *n; i++)
    {
        if (matri != vect[i].matri)
        {
            //Va metiendo los datos que no sean de la matricula buscada
            veck[k] = vect[i];
            k++;
        }
    }

    if (k == *n)
    {
        // Lo que hace es que si el for anterior se completo completamente es porque no se borro ninguna matricula y retorna el vector dado
        printf("MatrÃ cula no encontrada\n");
        return vect[N];
    }
    else
    {
        printf("Registro con matrÃ cula %d eliminado.\n", matri);
        *n = k;
        // Va actualizando el registro que le dimos porque eliminamos el registro
        for (int i = 0; i < k; i++)
        {
            vect[i] = veck[i];
        }
        //retorno ya el vector dados
        return vect[k];
    }
}

int genMatri(Registro vect[], int i)
{
    int num;
    do
    {
        num = (rand() % 100000) + 300000;
    } while (buscarAlumSec(vect, i, num) != -1);
    return num;
}

void ordenar_regis(Registro vect[], int n)
{
    int i, j;
    Registro temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (vect[j].matri < vect[i].matri)
            {
                temp = vect[i];
                vect[i] = vect[j];
                vect[j] = temp;
            }
        }
    }
}

int buscarAlumBin(Registro vect[], int i, int matricula)
{
    int izq, drcha;
    int med;
    izq = 0;
    drcha = i - 1;
    while (izq <= drcha)
    {
        med = izq + (drcha - izq) / 2;
        if (vect[med].matri == matricula)
        {
            return med;
        }

        if (vect[med].matri < matricula)
        {
            izq = med + 1;
        }
        else
        {
            drcha = med - 1;
        }
    }
    return -1;
}

int buscarAlumSec(Registro vect[], int i, int matricula)
{
    int j;
    for (j = 0; j < i; j++)
    {
        if (matricula == vect[j].matri)
        {
            return j;
        }
    }
    return -1;
}

void menuBusc(Registro vect[], int i, bool orden)
{

```

```

int pos;
if (orden)
{
    pos=buscarAlumBin(vect,i, vali_rango("Ingresa la matricula del alumno que desea buscar: ",300000,399999));
    if (pos!=-1)
    {
        printf("-----\n");
        printf("| \n");
        printf("| MATRIUCLA: %d \n", vect[i].matri);
        printf("| NOMBRE: %s \n", vect[i].persona.nombre1);
        printf("| SEGUNDO NOMBRE: %s \n", vect[i].persona.nombre2);
        printf("| APELLI PATERNO: %s \n", vect[i].persona.appater);
        printf("| APELLI MATERNO: %s \n", vect[i].persona.apmater);
        printf("| FECHA DE NACIM: %s-%s-%s \n", vect[i].nacimiento.dia, vect[i].nacimiento.mes, vect[i].nacimiento.anio);
        printf("| EDAD: %d \n", vect[i].edad);
        printf("| SEXO: %s \n", vect[i].sexo);
        printf("| LUGAR DE NACIM: %s \n", vect[i].estado_compl);
        printf("| CURP GENERADA : %s \n", vect[i].curp);
        printf("| \n");
        printf("-----\n");
    }
    else
    {
        printf("\nNo se encontro al alumno\n");
    }
}
else
{
    pos=buscarAlumSec(vect,i, vali_rango("Ingresa la matricula del alumno que desea buscar: ",300000,399999));
    if (pos!=-1)
    {
        printf("-----\n");
        printf("| \n");
        printf("| MATRIUCLA: %d \n", vect[i].matri);
        printf("| NOMBRE: %s \n", vect[i].persona.nombre1);
        printf("| SEGUNDO NOMBRE: %s \n", vect[i].persona.nombre2);
        printf("| APELLI PATERNO: %s \n", vect[i].persona.appater);
        printf("| APELLI MATERNO: %s \n", vect[i].persona.apmater);
        printf("| FECHA DE NACIM: %s-%s-%s \n", vect[i].nacimiento.dia, vect[i].nacimiento.mes, vect[i].nacimiento.anio);
        printf("| EDAD: %d \n", vect[i].edad);
        printf("| SEXO: %s \n", vect[i].sexo);
        printf("| LUGAR DE NACIM: %s \n", vect[i].estado_compl);
        printf("| CURP GENERADA : %s \n", vect[i].curp);
        printf("| \n");
        printf("-----\n");
    }
    else
    {
        printf("\nNo se encontro al alumno\n");
    }
}
}

int archivoTxt(Registro vect[],int i,int band)
{
    int j;
    FILE *pa;
    if (band)
    {
        pa=fopen("registros.txt","a");
        char str[140]={"| No | Est | Matri | Apellido Paterno | Apellido Materno | Nombre | Ed | Sex | CURP | \n"};
        fprintf(pa,"%s",str);

        for(j=0;j<i;j++)
        {
            if (vect[j].status)
            {
                fprintf(pa,"| %4d ",j);
                fprintf(pa,"| %s | %d | %-16s | %-17s | %-19s | %-19s | %3d | %c | ",vect[j].estado,vect[j].matri,vect[j].persona.appater,vect[j].persona.apmater,vect[j].persona.nombre1,vect[j].persona.nombre2,vect[j].nacimiento.dia,vect[j].nacimiento.mes,vect[j].nacimiento.anio,vect[j].edad,vect[j].sexo,vect[j].estado_compl,vect[j].curp);
                fprintf(pa,"%s | \n",vect[j].curp);
            }
        }
    }
    else
    {
        pa=fopen("registros.txt","w");
        char str[140]={"| No | Est | Matri | Apellido Paterno | Apellido Materno | Nombre | Ed | Sex | CURP | \n"};
        fprintf(pa,"%s",str);

        for(j=0;j<i;j++)
        {
            if (vect[j].status)
            {
                fprintf(pa,"| %4d ",j);
                fprintf(pa,"| %s | %d | %-16s | %-17s | %-19s | %-19s | %3d | %s | ",vect[j].estado,vect[j].matri,vect[j].persona.appater,vect[j].persona.apmater,vect[j].persona.nombre1,vect[j].persona.nombre2,vect[j].nacimiento.dia,vect[j].nacimiento.mes,vect[j].nacimiento.anio,vect[j].edad,vect[j].sexo,vect[j].estado_compl,vect[j].curp);
                fprintf(pa,"%s | \n",vect[j].curp);
            }
        }
    }

    fclose(pa);

    return 0;
}

```

```

//                               Ensenada, Baja California a 03 de Noviembre del 2023 //
// Nombre:                      Miguel Angel Ramirez Monjaraz //
// Matricula:                   372205 //
// Programa:                    Libreria de Gauss //
// Nombre del programa: MARM_ACT11_01
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

//definicion de funciones void//
void imprimir(char cadena[]);
void a_mayusculas(char cadena[]);
void vocales_conso(char cadena[],char vocal[],char consonate[]);
void separar_nombre(char cadena[]);
void validar_datos(char cadena[]);
void eliminarEspacioInicial(char cadena[]);

//definicion de funcion int //
int letras_simbolos(char cadena[]);
int tam_carac(char cadena[]);
int vali_rango(char msge[],int ri,int rf);

int vali_rango(char msge[],int ri,int rf)
{
    int num;
    char xnum[30];
    do
    {
        puts(msge);
        fflush(stdin);
        gets(xnum);
        num = atoi(xnum);
    }while(num < ri || num > rf);
    return num;
}

int tam_carac(char cadena[])
{
    int i;
    for(i=0;cadena[i] != '\0';i++);
    return i;
}

int letras_simbolos(char caracter[])
{
    for (int i = 0; caracter[i] != '\0'; i++) {
        if (!((caracter[i] >= 65 && caracter[i] <= 90) || // Letras mayÃsculas
            (caracter[i] >= 97 && caracter[i] <= 122) || // Letras minÃsculas
            (caracter[i] == 39 || caracter[i] == 44 || // Caracteres permitidos
            (caracter[i] >= 45 && caracter[i] <= 47) ||
            (caracter[i] == 95 || caracter[i] == 96) ||
            caracter[i] == 32))) // Espacio
        {
            return 0;
        }
        else
        {
            if(caracter[0] == 32)
            {
                return 0;
            }
            if (caracter[i - 1] == 32 && caracter[i] == 32)
            {
                return 0;
            }
        }
    }
    return 1;
}

void validar_datos(char msge[],char cadena[])
{
    int i=1;
    do
    {
        puts(msge);
        fflush(stdin);
        gets(cadena);
        if(letras_simbolos(cadena))
        {
            i=0;
        }
    } while (i == 1);
}

void a_mayusculas(char cadena[])
{
    int i;
    for(i=0;cadena[i] != '\0';i++)
    {
        if(cadena[i] >= 97)
        {
            if(cadena[i] <= 122)

```

```

        {
            cadena[i] = cadena[i] - 32;
        }
    }
}

void vocales_conso(char cadena[],char vocal[],char consonante[])
{
    int j,i=0,k=0;
    for(j=0;cadena[j] != '\0';j++)
    {
        if (cadena[j] != 'A' && cadena[j] != 'E' && cadena[j] != 'I' && cadena[j] != 'O' && cadena[j] != 'U')
        {
            consonante[i] = cadena[j];
            i++;
        }
        else
        {
            //VOCALES//
            vocal[k] = cadena[j];
            k++;
        }
    }
    consonante[i] = '\0';
    vocal[i] = '\0';
}

void separar_nombre(char cadena[])
{
    char compues_prohi[20][8] = {"DA", "DAS", "DE", "DEL", "DER", "DI", "DIE", "DD", "Y", "EL", "LA", "LOS", "LAS", "LE", "LES", "MAC", "MC", "VAN", "VON"};
    char nueva_cadena[100];
    int i = 0;
    int j = 0;
    int longitud = tam_carac(cadena);

    while (i < longitud) {
        int longitud_palabra = 0;
        while (i < longitud && cadena[i] != ' ') {
            longitud_palabra++;
            i++;
        }

        char palabra[15];
        strncpy(palabra, &cadena[i - longitud_palabra], longitud_palabra);
        palabra[longitud_palabra] = '\0';

        int es_compuesto = 0;
        for (int k = 0; k < 20; k++) {
            if (strcmp(palabra, compues_prohi[k]) == 0) {
                es_compuesto = 1;
                break;
            }
        }

        if (!es_compuesto) {
            for (int k = 0; k < longitud_palabra; k++) {
                nueva_cadena[j] = palabra[k];
                j++;
            }
            nueva_cadena[j] = ' ';
            j++;
        }
        while (i < longitud && cadena[i] == ' ') {
            i++;
        }
    }

    nueva_cadena[j] = '\0';
    strcpy(cadena, nueva_cadena);
}

void quitar_ene(char cadena[])
{
    int i,longi;
    longi = tam_carac(cadena);
    for(i = 0 ;i<longi;i++)
    {
        if(cadena[i] == -92)
        {
            {
                cadena[i] = 'X';
            }
        }
        if(cadena[i] == -91)
        {
            {
                cadena[i] = 'X';
            }
        }
    }
}

void eliminarEspacioInicial(char cadena[])
{
    if (cadena[0] == ' ')
    {
        int longitud = tam_carac(cadena);
        for (int i = 0; i < longitud; i++)
        {
            {
                cadena[i] = cadena[i + 1];
            }
        }
    }
}

```



```

//                               Ensenada, Baja California a 03 de Noviembre del 2023 //
// Nombre:                       Miguel Angel Ramirez Monjaraz //
// Matricula:                     372205 //
// Programa:                      Libreria que genera curp //
// Nombre del programa: MARM_ACT11_01 //

#include "Gauss.h"
char lugar(char estado[]);

char composicion[81][5] = {"BAKA", "BACA", "BUEI", "BUEY", "CACA", "CACO", "CAGA", "CAGO", "CAKA", "CAKO", "COGE",
                           "COGI", "COJA", "COJE", "COJI", "COJO", "COLA", "CULO", "FALO", "FETO", "GETA", "GUEI", "GUEY", "JETA", "JOTO", "KACA",
                           "KACO", "KAGA", "KAGO", "KAKA", "KAKO", "KOGA", "KOGI", "KOJA", "KOJE", "KOJI", "KOJO", "KOLA", "KULO", "LILO", "LOCA", "LOCO", "LOKA", "LOKO",
                           "MAME", "MAMO", "MEAR", "MEAS", "MEON", "MIAR", "MION", "MOCO", "MOKO", "MULA", "MULO", "NACA", "NACO", "PEDA", "PEDO", "PENE", "PIPI", "PITO",
                           "POPO", "PUTA", "PUTO", "QULO", "RATA", "ROBA", "ROBE", "ROBO", "RUIN", "SENO", "TETA", "VACA", "VAGO", "VAGA", "VAKA", "VUEI", "WUEI", "WUEY"};

char nombres_prohibidos[10][8] = {"MARIA", "MA.", "MA", "M.", "M", "JOSE", "J.", "J"};
char letras_feas[6][4] = {"A", "N", "L", "S", "R", "T"};

void generar_curp(char CURP[], char nombr1[], char nombr2[], char apelli1[], char apelli2[], char dia[], char mes[], char anio[], char sex[], char estado[])
{
    int i, j, k, op1, op2, op3, sin_nombre = 0, usar_nom2 = 0, num, num2;
    char numc[3], num2c[3];

    char nombr1_vocal[20];
    char nombr2_vocal[20];
    char apelli1_vocal[30];
    char apelli2_vocal[30];

    char nombr1_conso[20];
    char nombr2_conso[20];
    char apelli1_conso[30];
    char apelli2_conso[30];

    eliminarEspacioInicial(nombr1);
    eliminarEspacioInicial(nombr2);
    eliminarEspacioInicial(apelli1);
    eliminarEspacioInicial(apelli2);

    a_mayusculas(nombr1);
    a_mayusculas(nombr2);
    a_mayusculas(apelli1);
    a_mayusculas(apelli2);
    quitar_ene(nombr1);
    quitar_ene(nombr2);
    quitar_ene(apelli1);
    quitar_ene(apelli2);

    for (int i = 0; i < 10; i++)
    {
        if (strcmp(nombr1, nombres_prohibidos[i]) == 0)
        {
            usar_nom2 = 1;
            break;
        }
    }
    // aqui comprobamos si trae mas de una palabra sus nombre y apellidos
    separar_nombre(nombr1);
    separar_nombre(nombr2);
    separar_nombre(apelli1);
    separar_nombre(apelli2);

    //separamos es vocales y consonantes para chambear mejorcito
    vocales_conso(nombr1, nombr1_vocal, nombr1_conso);
    vocales_conso(nombr2, nombr2_vocal, nombr2_conso);
    vocales_conso(apelli1, apelli1_vocal, apelli1_conso);
    vocales_conso(apelli2, apelli2_vocal, apelli2_conso);
    if (tam_carac(nombr2) > 0)
    {
        op1 = 1;
    }
    if (tam_carac(apelli1) > 0)
    {
        op2 = 1;
    }
    if (tam_carac(apelli2) > 0)
    {
        op3 = 1;
    }

    if (op2 == 1)
    {
        // tiene primer apellido
        CURP[0] = apelli1[0];
        if (apelli1[0] == 'A' || apelli1[0] == 'E' || apelli1[0] == 'I' || apelli1[0] == 'O' || apelli1[0] == 'U')
        {
            CURP[1] = apelli1_vocal[1];
        }
        else
        {
            CURP[1] = apelli1_vocal[0];
        }
    }
    else
    {
        // sin primer apellido
        CURP[0] = 'X';
        CURP[1] = 'X';
        sin_nombre = 1;
    }
    if (op3 == 1)
    {
        CURP[2] = apelli2[0];
    }
}

```

```

else
{
    CURP[2] = 'X';
}

for (int i = 0; i < 10; i++)
{
    if (strcmp(nombr1, nombres_prohibidos[i]) == 0)
    {
        usar_nom2 = 1;
        break;
    }
}
if (usar_nom2 != 1)
{
    CURP[3] = nombr1[0];
}
else
{
    CURP[3] = nombr2[0];
}
// checamos si las primeras cuatro letras forman la palabra prohibida//
for (int i = 0; i < 81; i++)
{
    if (strcmp(CURP, composicion[i]) == 0)
    {
        CURP[1] = 'X';
        break;
    }
}

CURP[4] = anio[2];
CURP[5] = anio[3];
CURP[6] = mes[0];
CURP[7] = mes[1];
CURP[8] = dia[0];
CURP[9] = dia[1];
CURP[10] = sex[0];
CURP[11] = estado[0];
CURP[12] = estado[1];
if (op2 == 1)
{
    if (CURP[0] == apelli1_conso[0])
    {
        if (strlen(apelli1_conso) > 2)
        {
            CURP[13] = apelli1_conso[1];
        }
        else
        {
            CURP[13] = 'X';
        }
    }
    else
    {
        CURP[13] = apelli1_conso[0];
    }
}
else
{
    CURP[13] = 'X';
}

if (op3 == 1)
{
    if (strlen(apelli2_conso) > 2)
    {
        if (CURP[2] == apelli2_conso[0])
        {
            CURP[14] = apelli2_conso[1];
        }
        else
        {
            CURP[14] = apelli2_conso[0];
        }
    }
    else
    {
        CURP[14] = 'X';
    }
}
else
{
    CURP[14] = 'X';
}
for (int i = 0; i < 10; i++)
{
    if (strcmp(nombr1, nombres_prohibidos[i]) == 0)
    {
        usar_nom2 = 1;
        break;
    }
}
if (usar_nom2 != 1)
{
    if (CURP[3] == nombr1_conso[0])
    {
        if (strlen(nombr1_conso) > 2)
        {
            CURP[15] = nombr1_conso[1];
        }
        else
        {
            CURP[15] = 'X';
        }
    }
}

```

```

    }
    else
    {
        CURP[15] = nombr1_conso[0];
    }
}

else
{
    if(CURP[3] == nombr2_conso[0])
    {
        if(strlen(nombr2_conso) > 2)
        {
            CURP[15] = nombr2_conso[1];
        }
        else
        {
            CURP[15] = 'X';
        }
    }
    else
    {
        CURP[15] = nombr2_conso[0];
    }
}

if (anio < "1999")
{
    num = rand() % 9;
    sprintf(numc, "%d", num);
    CURP[16] = numc[0];
}

else
{
    if (anio <= "2010")
    {
        CURP[16] = 'A';
    }
    else
    {
        if (anio <= "2020")
        {
            CURP[16] = 'B';
        }
        else
        {
            if (anio <= "2030")
            {
                CURP[16] = 'C';
            }
        }
    }
}

}

void genEst(char estado[],char estado_comp[])
{
    int E;
    char estList[33][3]={
        "AS","BC","BS","CC","CL","CM","CS", "CH","DE","DG",
        "GT","GR","HG","JC","MC","MN","MS","NT","NL","OC",
        "PL","QT","QR","SP","SL","SR","TC","TS","TL","VZ",
        "YN","ZS","NE"
    };
    char estlist_com[33][20]={
        "AGUASCALIENTES","BAJA CALIFORNIA","BAJA CALIFORNIA SUR", "CAMPECHE","COAHUILA","COLIMA","CHIAPAS",
        "CHIHUAHUA","DISTRITO FEDERAL","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"
    };

    E=rand()%33;

    estado[0]=estList[E][0];
    estado[1]=estList[E][1];
    strcpy(estado_comp, estlist_com[E]);
    estado[2]='\0';
}

// FunciÃ³n para determinar si un aÃ±o es bisiesto
int esBisiesto(int year)
{
    if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
    {
        return 1; // Es bisiesto
    } else {
        return 0; // No es bisiesto
    }
}

int randomDia(int mes, int bisiesto)
{
    int days;
    switch (mes) {
        case 4:
        case 6:
        case 9:
        case 11:
            days = 30;
            break;
    }
}

```

```

        case 2:
            if (bisiesto)
                days = 29;
            else
                days = 28;
            break;
        default:
            days = 31;
            break;
    }
    return rand() % days + 1;
}

int calcularEdad(char anio[], char mes[], char dia[])
{
    int anio, mes, dia;
    anio = atoi(anio);
    mes = atoi(mes);
    dia = atoi(dia);

    time_t now = time(0);
    struct tm tm = *localtime(&now);

    int edad = tm.tm_year + 1900 - anio;

    if (tm.tm_mon + 1 < mes || (tm.tm_mon + 1 == mes && tm.tm_mday < dia))
    {
        edad--;
    }

    return edad;
}

void generarFechaAlea(char anio[], char mes[], char dia[])
{
    int ri=1900, rf=2023, rango=(rf-ri+1);
    int day;
    int year = rand() % (2023 - 1930 + 1) + 1930;
    int month = rand() % 12 + 1;
    if (esBisiesto(year) == 1)
    {
        day = randomDia(month, 1);
    }
    else
    {
        day = randomDia(month, 1);
    }
    char aux;
    itoa(year, anio, 10);
    itoa(month, mes, 10);
    itoa(day, dia, 10);
    if (strlen(mes) == 1)
    {
        aux = mes[0];
        mes[0] = '0';
        mes[1] = aux;
    }

    if (strlen(dia) == 1)
    {
        aux = dia[0];
        dia[0] = '0';
        dia[1] = aux;
    }
}

```