

```

/* Mej   a Mart   nez, Ram  n      372099      */
/* 22 de octubre de 2023      */
/* Implementacion de la curp      */
/* MMR_ACT9_3.4_932      */

#include "C:\Users\ramon\OneDrive\Escritorio\MMR_ACT1_932\MMRLIB.h"

#define STRING 100

void validStrCurp(char str[]);
int datos(char name[],char appat[],char apmat[],char anio[],char mes[],char dia[],char estado[]);
void fecha(char anio[],char mes[],char dia[]);
void Est(char estado[]);
int diaValid(int mes,bool leap);
char vocalInterna(char appat[]);
char consInterna(char str[]);
int dieresis(char c);
void validCurp(char curp[],char name[],char appat[],char apmat[]);
void validName(char curp[],char name[],char appat[],char apmat[]);
int compName(char name[]);
int compAp(char ap[]);
int espacio(char str[]);
void curp(void);
char curpl6(char anio[]);
int simbolo(char c1);
void validSimbolo(char curp[]);
int validPalabraInconven(char curp[]);

int main()
{
    int op;
    do
    {
        system("cls");
        printf("Curp\n\n");
        printf("1) Generar curp\n");
        printf("2) Salir\n");
        op=validNum("Escoge una opcion: ",1,2);
        if(op==1)
        {
            curp();
        }
        printf("\n");
        system("pause");
    }while(op!=2);
    return 0;
}

int diaValid(int mes,bool leap)
{
    switch(mes)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            return 31;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            return 30;
            break;
        case 2:
            if(leap==true)
            {
                return 29;
            }
            else
            {
                return 28;
            }
        }
    }

```

```

        }
        break;
    }
}

int datos(char name[],char appat[],char apmat[],char anio[],char mes[],char dia[],char estado[])
{
    int nap;
    int sex;
    system("cls");
    printf(" Datos\n\n");
    printf("Ingresa tu nombre: ");
    validStrCurp(name);
    strM(name);

    printf("\nTiene apellido paterno?\n");
    printf("1) Si\n");
    printf("2) No\n");
    nap=validNum("Escoge una opcion: ",1,2);
    if(nap==1)
    {
        printf("\nIngresa tu apelliddo paterno: ");
        validStrCurp(appat);
        strM(appat);
    }
    else
    {
        appat[0]='X';
        appat[1]='\0';
    }

    printf("\nTiene apellido materno?\n");
    printf("1) Si\n");
    printf("2) No\n");
    nap=validNum("Escoge una opcion: ",1,2);
    if(nap==1)
    {
        printf("\nIngresa tu apelliddo materno: ");
        validStrCurp(apmat);
        strM(apmat);
    }
    else
    {
        apmat[0]='X';
        apmat[1]='\0';
    }

    printf("\n");
    system("pause");
    system("cls");
    printf("Fecha de nacimiento\n\n");
    fecha(anio,mes,dia);

    printf("\n");
    system("pause");
    Est(estado);

    printf("\n");
    system("pause");
    system("cls");
    sex=validNum("\nIngresa tu sexo\n1) Masculino\n2) Femenino\nElige una opcion: ",1,2);

    return sex;
}

void curp(void)
{
    srand(time(NULL));
    //datos
    char rnd[2];
    char curp[19];
    char name[STRING], appat[STRING], apmat[STRING];
    char anio[4],mes[2],dia[2],estado[2],sex;
    char s;

```

```

s=datos(name, appat, apmat, anio, mes, dia, estado);
if (s==1)
{
    sex='H';
}
else
{
    sex='M';
}

curp[0]=appat[0];
curp[1]=vocalInterna(appat);
curp[2]=apmat[0];
curp[3]=name[0];
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;
curp[11]=estado[0];
curp[12]=estado[1];
curp[13]=consInterna(appat);
curp[14]=consInterna(apmat);
curp[15]=consInterna(name);
curp[16]=curp16(anio);
itoa(rand()%10, rnd, 10);
curp[17]=rnd[0];
curp[18]='\0';

/*
system("cls");
printf("\n");
printf("Curp: ");
strprint(curp);
*/
validCurp(curp, name, appat, apmat);

system("cls");
printf("Curp: ");
printf("%s", curp);
}

void fecha(char anio[], char mes[], char dia[])
{
    int born[3];
    bool leap=false;
    born[2]=validNum("\nIngresa tu aÃ±o de nacimiento: ", 1900, 2023);
    itoa(born[2], anio, 10);

    born[1]=validNum("\nIngresa tu mes de nacimiento: ", 1, 12);
    itoa(born[1], mes, 10);
    if(born[1]>=10)
    {
        itoa(born[1], mes, 10);
    }
    else
    {
        itoa(born[1], mes, 10);
        mes[1]=mes[0];
        mes[0]='\0';
    }

    if(born[1]==2) //mes
    {
        if(born[2]%4==0) //aÃ±o
        {
            leap=true;
        }
    }

    born[0]=validNum("\nIngresa tu dia de nacimiento: ", 1, diaValid(born[1], leap));
    if(born[0]>=10)
    {
        itoa(born[0], dia, 10);
    }
}

```

```

    }
    else
    {
        itoa(born[0], dia, 10);
        dia[1] = dia[0];
        dia[0] = '0';
    }
}

char vocalInterna(char appat[])
{
    int i;
    for(i=1; appat[i] != '\0'; i++)
    {
        if(simbolo(appat[i]))
        {
            return appat[i];
        }
        if(appat[i] == 'A')
        {
            return appat[i];
        }
        if(appat[i] == 'E')
        {
            return appat[i];
        }
        if(appat[i] == 'I')
        {
            return appat[i];
        }
        if(appat[i] == 'O')
        {
            return appat[i];
        }
        if(appat[i] == 'U')
        {
            return appat[i];
        }

        if(dieresis(appat[i]))
        {
            return 'X';
        }
    }
    return 'X';
}

char consInterna(char str[])
{
    int i;
    for(i=1; str[i] != '\0'; i++)
    {
        if(str[i] != 'A')
        {
            if(str[i] != 'E')
            {
                if(str[i] != 'I')
                {
                    if(str[i] != 'O')
                    {
                        if(str[i] != 'U')
                        {
                            if(!dieresis(str[i]))
                            {
                                return str[i];
                            }
                        }
                    }
                }
            }
        }
    }
    return 'X';
}

```

```

void Est(char estado[])
{
    int E;
    char estList[33][3]={
        "AS","BC","BS","CC","CS","CH","CM","CL","CM","DG",
        "GT","GR","HG","JC","EM","MN","MS","NT","NL","OC",
        "PL","QT","QR","SP","SL","SR","TC","TS","TL","VZ",
        "YN","ZS","NE"
    };

    system("cls");
    printf("Estado\n\n");
    printf("1) Aguascalientes\n");
    printf("2) Baja California\n");
    printf("3) Baja California Sur\n");
    printf("4) Campeche\n");
    printf("5) Chiapas\n");
    printf("6) Chihuahua\n");
    printf("7) Ciudad de MÃ©xico\n");
    printf("8) Coahuila\n");
    printf("9) Colima\n");
    printf("10) Durango\n");
    printf("11) Guanajuato\n");
    printf("12) Guerrero\n");
    printf("13) Hidalgo\n");
    printf("14) Jalisco\n");
    printf("15) Estado de MÃ©xico\n");
    printf("16) MichoacÃ¡n\n");
    printf("17) Morelos\n");
    printf("18) Nayarit\n");
    printf("19) Nuevo LeÃ³n\n");
    printf("20) Oaxaca\n");
    printf("21) Puebla\n");
    printf("22) QuerÃ©taro\n");
    printf("23) Quintana Roo\n");
    printf("24) San Luis PotosÃ  \n");
    printf("25) Sinaloa\n");
    printf("26) Sonora\n");
    printf("27) Tabasco\n");
    printf("28) Tamaulipas\n");
    printf("29) Tlaxcala\n");
    printf("30) Veracruz\n");
    printf("31) YucatÃ¡n\n");
    printf("32) Zacatecas\n");
    printf("33) Extranjero\n");
    E=validNum("Escoge tu estado: ",1,33);

    estado[0]=estList[E-1][0];
    estado[1]=estList[E-1][1];
}

```

```

void validStrCarp(char str[])
{
    int i,valido,band,espacio;
    do
    {
        fflush(stdin);
        gets(str);
        band=1;
        espacio=1;
        i=0;
        do
        {
            valido=0;
            //fflush(stdin);
            //c=getch();
            if(str[i]>='A')
            {
                if(str[i]<='Z')
                {
                    valido++;
                }
            }

            if(str[i]>='a')

```

```

{
    if(str[i]<='z')
    {
        valido++;
    }
}

if(str[i]==-91) //Ã±
{
    str[i]='X';
    valido++;
}

if(str[i]==-92) //Ä
{
    str[i]='X';
    valido++;
}

if(simbolo(str[i]))
{
    valido++;
}

if(dieresis(str[i]))
{
    valido++;
}

if(str[i]==-127)
{
    str[i]='U';
    valido++;
}
if(str[i]==-102)
{
    str[i]='U';
    valido++;
}
if(str[i]==' ')
{
    if(i==0)
    {
        printf("\nla cadena no puede iniciar con espacio\n");
        printf("\nVuela a ingresar la cadena: ");
        espacio=0;
    }
    else
    {
        if(str[i-1]==' ')
        {
            printf("\nla cadena no tener dos espacios consecutivos\n");
            printf("\nVuela a ingresar la cadena: ");
            espacio=0;
        }
        else
        {
            valido++;
        }
    }
}

if(str[i]=='\0')
{
    if(str[i-1]!=' ')
    {
        valido++;
        band=0;
    }
    else
    {
        printf("\nla cadena no puede acabar con espacio\n");
        printf("\nVuela a ingresar la cadena: ");
    }
}

```

```

        espacio=0;
    }
}
if(valido!=0)
{
    i++;
}
else
{
    if(espacio)
    {
        printf("\nla cadena tiene caracteres invalidos\n");
        printf("\nVuela a ingresar la cadena: ");
    }
    band=0;
}
}while(band);
}while(valido==0);
}

char curp16(char anio[])
{
    if(anio[0]=='1')
    {
        return anio[2];
    }
    else
    {
        switch(anio[2])
        {
            case '0':
                return 'A';
                break;
            case '1':
                return 'B';
                break;
            case '2':
                return 'C';
                break;
        }
    }
}

int dieresis(char c)
{
    switch(c)
    {
        case -114: //a
        case -45:  //e
        case -40:  //i
        case -103: //o
        case -124: //a
        case -119: //e
        case -117: //i
        case -108: //o
            return 1;
            break;
        default:
            return 0;
    }
}

void validCurp(char curp[],char name[],char appat[],char apmat[])
{
    validName(curp,name,appat,apmat);

    //dieresis
    if(dieresis(curp[0]))
    {
        curp[0]='X';
    }
    if(dieresis(curp[2]))
    {
        curp[2]='X';
    }
}

```

```

    }
    if(dieresis(curp[3]))
    {
        curp[3]='X';
    }
    system("pause");

    validSimbolo(curp);

    if(validPalabraInconven(curp))
    {
        curp[1]='X';
    }
}

void validName(char curp[],char name[],char appat[],char apmat[])
{
    int i,j,k;
    int r;
    char name2[STRING];
    char name3[STRING];
    char name4[STRING];
    char appat2[STRING];
    char appat3[STRING];
    char appat4[STRING];
    char apmat2[STRING];
    char apmat3[STRING];
    char apmat4[STRING];

    if(espacio(name))
    {
        r=3;
        if(compName(name))
        {
            i=0;
            while(name[i]!=' ')
            {
                i++;
            }
            i++;
            k=0;
            for(j=i;name[j]!='\0';j++)
            {
                name2[k]=name[j];
                k++;
            }
            name2[k]='\0';
            r=0;
        }
        if(espacio(name2))
        {
            if(compName(name2))
            {
                i=0;
                while(name2[i]!=' ')
                {
                    i++;
                }
                i++;
                k=0;
                for(j=i;name2[j]!='\0';j++)
                {
                    name3[k]=name2[j];
                    k++;
                }
                name3[k]='\0';
                r=1;
            }
        }
        if(espacio(name3))
        {
            if(compName(name3))
            {
                i=0;

```



```

        while (name3[i] != ' ')
        {
            i++;
        }
        i++;
        k=0;
        for (j=i; name3[j] != '\0'; j++)
        {
            name4[k]=name3[j];
            k++;
        }
        name4[k]='\0';
        r=2;
    }
}

if (r==0)
{
    curp[3]=name2[0];
    curp[15]=consInterna (name2);
}
else
{
    if (r==1)
    {
        curp[3]=name3[0];
        curp[15]=consInterna (name3);
    }
    else
    {
        if (r==2)
        {
            curp[3]=name4[0];
            curp[15]=consInterna (name4);
        }
    }
}
}

if (espacio (appat))
{
    r=3;
    if (compAp (appat))
    {
        i=0;
        while (appat[i] != ' ')
        {
            i++;
        }
        i++;
        k=0;
        for (j=i; appat[j] != '\0'; j++)
        {
            appat2[k]=appat[j];
            k++;
        }
        appat2[k]='\0';
        r=0;
    }
    if (espacio (appat2))
    {
        if (compName (appat2))
        {
            i=0;
            while (appat2[i] != ' ')
            {
                i++;
            }
            i++;
            k=0;
            for (j=i; appat2[j] != '\0'; j++)
            {
                appat3[k]=appat2[j];
                k++;
            }
        }
    }
}

```

```

    }
    appat3[k]='\0';
    r=1;
}
}
if(espacio(appat3))
{
    if(compName(appat3))
    {
        i=0;
        while(appat3[i]!=' ')
        {
            i++;
        }
        i++;
        k=0;
        for(j=i;appat3[j]!='\0';j++)
        {
            appat4[k]=appat3[j];
            k++;
        }
        appat4[k]='\0';
        r=2;
    }
}
if(r==0)
{
    curp[0]=appat2[0];
    curp[1]=vocalInterna(appat2);
    curp[13]=consInterna(appat2);
}
else
{
    if(r==1)
    {
        curp[0]=appat3[0];
        curp[1]=vocalInterna(appat3);
        curp[13]=consInterna(appat3);
    }
    else
    {
        if(r==2)
        {
            curp[0]=appat4[0];
            curp[1]=vocalInterna(appat4);
            curp[13]=consInterna(appat4);
        }
    }
}
}
if(espacio(apmat))
{
    r=3;
    if(compAp(apmat))
    {
        i=0;
        while(apmat[i]!=' ')
        {
            i++;
        }
        i++;
        k=0;
        for(j=i;apmat[j]!='\0';j++)
        {
            apmat2[k]=apmat[j];
            k++;
        }
        apmat2[k]='\0';
        r=0;
    }
}
if(espacio(apmat2))
{
    if(compName(apmat2))
    {
        i=0;

```

```

        while (apmat2[i] != ' ')
        {
            i++;
        }
        i++;
        k=0;
        for (j=i; apmat2[j] != '\0'; j++)
        {
            apmat3[k]=apmat2[j];
            k++;
        }
        apmat3[k]='\0';
        r=1;
    }
}
if (espacio (apmat3))
{
    if (compName (apmat3))
    {
        i=0;
        while (apmat3[i] != ' ')
        {
            i++;
        }
        i++;
        k=0;
        for (j=i; apmat3[j] != '\0'; j++)
        {
            apmat4[k]=apmat3[j];
            k++;
        }
        apmat4[k]='\0';
        r=2;
    }
}
if (r==0)
{
    curp[2]=apmat2[0];
    curp[14]=consInterna (apmat2);
}
else
{
    if (r==1)
    {
        curp[2]=apmat3[0];
        curp[14]=consInterna (apmat3);
    }
    else
    {
        if (r==2)
        {
            curp[2]=apmat4[0];
            curp[14]=consInterna (apmat4);
        }
    }
}
}
}

int compName(char name[])
{
    int i,j,suma;
    char no_validName[27][strlen(name)]={"MARIA", "MA", "MA.", "M.", "M", "JOSE", "J.", "J", "DA", "DAS", "DE", "DEL", "DER",
        "DI", "DIE", "DD", "Y", "EL", "LA", "LOS", "LAS", "LE", "LES", "MAC", "MC", "VAN", "VON"};

    for (j=0; j<27; j++)
    {
        suma=0;
        for (i=0; name[i] != ' '; i++)
        {
            if (name[i]==no_validName[j][i])
            {

```

```

        suma++;
    }
}

if(suma==i)
{
    suma++;
    return 1;
}
return 0;
}

int compAp(char ap[])
{
    int i,j,suma=0;
    char no_validAp[19][strlen(ap)]={
        "DA","DAS","DE","DEL","DER","DI","DIE","DD","Y",
        "EL","LA","LOS","LAS","LE","LES","MAC","MC","VAN",
        "VON"
    };

    for(j=0;j<19;j++)
    {
        suma=0;
        for(i=0;ap[i]!='\0';i++)
        {
            if(ap[i]==no_validAp[j][i])
            {
                suma++;
            }
        }

        if(suma==i)
        {
            suma++;
            return 1;
        }
    }
    return 0;
}

int simbolo(char c1)
{
    switch(c1)
    {
        case 39:
        case 45:
        case 47:
        case 46:
            return 1;
            break;
        default:
            return 0;
    }
}

int espacio(char str[])
{
    int i;
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]==' ')
        {
            return 1;
        }
    }
    return 0;
}

void validSimbolo(char curp[])
{
    int i;
    for(i=0;curp[i]!='\0';i++)

```

```

    {
        if(simbolo(curp[i]))
        {
            curp[i]='X';
        }
    }
}

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

    for(i=0;i<81;i++)
    {
        suma=0;
        for(j=0;j<4;j++)
        {
            if(curp[j]==inconv[i][j])
            {
                suma++;
            }
        }
        if(suma==j)
        {
            return 1;
        }
    }
    return 0;
}

```

Pruebas de escritorio:

Menu:

```
Curp  
  
1) Generar curp  
2) Salir  
Escoge una opcion: █
```

Nombre:

```
Datos  
  
Ingresa tu nombre: ramon
```

Apellido paterno:

```
Tiene apellido paterno?  
1) Si  
2) No  
Escoge una opcion: 1  
  
Ingresa tu apelliddo paterno: mejia█
```

Apellido materno:

```
Tiene apellido materno?  
1) Si  
2) No  
Escoge una opcion: 1  
  
Ingresa tu apelliddo materno: martinez  
  
Presione una tecla para continuar . . . █
```

Fecha de nacimiento:

Fecha de nacimiento

Ingresa tu año de nacimiento: 2003

Ingresa tu mes de nacimiento: 2

Ingresa tu día de nacimiento: 24

Presione una tecla para continuar . . .

Estado:

Estado

- 1) Aguascalientes
- 2) Baja California
- 3) Baja California Sur
- 4) Campeche
- 5) Chiapas
- 6) Chihuahua
- 7) Ciudad de México
- 8) Coahuila
- 9) Colima

- 30) Veracruz
 - 31) Yucatán
 - 32) Zacatecas
 - 33) Extranjero
- Escoge tu estado: 2

Presione una tecla para continuar . . .

Sexo:

Ingresa tu sexo

- 1) Masculino
- 2) Femenino

Elige una opción: 1

Presione una tecla para continuar . . .

Curp:

Curp: MEMR030224HBCJRMA5

Presione una tecla para continuar . . . █