**Sergio Moncada Muñoz**

**1089380570**

**Programación 2 – Grupo 4**

**Proyecto final**

**Uso de inteligencia artificial (ChatGPT) para el desarrollo del proyecto**

funcion que reciba un numero y lo escriba en letras

void escribir\_numero\_en\_letras(long numero, char\* resultado, int tamano) {

const char\* unidades[] = {"cero", "uno", "dos", "tres", "cuatro", "cinco", "seis", "siete", "ocho", "nueve"};

const char\* especiales[] = {"diez", "once", "doce", "trece", "catorce", "quince", "dieciséis", "diecisiete", "dieciocho", "diecinueve"};

const char\* decenas[] = {"", "", "veinte", "treinta", "cuarenta", "cincuenta", "sesenta", "setenta", "ochenta", "noventa"};

const char\* centenas[] = {"", "ciento", "doscientos", "trescientos", "cuatrocientos", "quinientos", "seiscientos", "setecientos", "ochocientos", "novecientos"};

const char\* miles[] = {"", "mil", "millon", "millones"};

if (numero < 0 || numero > 999999999) {

strncpy(resultado, "Numero fuera de rango", tamano);

return;

}

if (numero < 10) {

strncpy(resultado, unidades[numero], tamano);

} else if (numero < 20) {

strncpy(resultado, especiales[numero - 10], tamano);

} else if (numero < 100) {

int decena = numero / 10;

int unidad = numero % 10;

if (unidad == 0) {

strncpy(resultado, decenas[decena], tamano);

} else {

snprintf(resultado, tamano, "%s y %s", decenas[decena], unidades[unidad]);

}

} else if (numero < 1000) {

int centena = numero / 100;

int decena = (numero % 100) / 10;

int unidad = numero % 10;

if (decena == 0 && unidad == 0) {

strncpy(resultado, centenas[centena], tamano);

} else {

char temp\_resultado[100];

escribir\_numero\_en\_letras(numero % 100, temp\_resultado, sizeof(temp\_resultado));

snprintf(resultado, tamano, "%s %s", centenas[centena], temp\_resultado);

}

} else if (numero < 1000000) {

int mil = numero / 1000;

int centena = (numero % 1000) / 100;

int decena = (numero % 100) / 10;

int unidad = numero % 10;

if (mil == 1 && centena == 0 && decena == 0 && unidad == 0) {

strncpy(resultado, "mil", tamano);

} else {

char temp\_resultado1[100];

char temp\_resultado2[100];

escribir\_numero\_en\_letras(mil, temp\_resultado1, sizeof(temp\_resultado1));

if (numero % 1000 == 0) {

snprintf(resultado, tamano, "%s mil", temp\_resultado1);

} else {

escribir\_numero\_en\_letras(numero % 1000, temp\_resultado2, sizeof(temp\_resultado2));

snprintf(resultado, tamano, "%s mil %s", temp\_resultado1, temp\_resultado2);

}

}

} else if (numero < 1000000000) {

int millon = numero / 1000000;

int mil = (numero % 1000000) / 1000;

int centena = (numero % 1000) / 100;

int decena = (numero % 100) / 10;

int unidad = numero % 10;

if (millon == 1 && mil == 0 && centena == 0 && decena == 0 && unidad == 0) {

strncpy(resultado, "un millon", tamano);

} else {

char temp\_resultado1[100];

char temp\_resultado2[100];

escribir\_numero\_en\_letras(millon, temp\_resultado1, sizeof(temp\_resultado1));

if (numero % 1000000 == 0) {

snprintf(resultado, tamano, "%s millones", temp\_resultado1);

} else {

escribir\_numero\_en\_letras(numero % 1000000, temp\_resultado2, sizeof(temp\_resultado2));

snprintf(resultado, tamano, "%s millones %s", temp\_resultado1, temp\_resultado2);

}

}

} else {

strncpy(resultado, "Numero fuera de rango", tamano);

}

}

Haz una función que valide el formato correcto de una fecha DD-MM-AAAA, (la recibe como cadena de caracteres)

bool fecha\_valida (const char fecha [])

{

int dia = (fecha[0] - '0') \* 10 + (fecha[1] - '0');

int mes = (fecha[3] - '0') \* 10 + (fecha[4] - '0');

int year = (fecha[6] - '0') \* 1000 + (fecha[7] - '0') \* 100 + (fecha[8] - '0') \* 10 + (fecha[9] - '0');

if (dia < 1 || dia > 31 || mes < 1 || mes > 12 || year < 0 || year >2023) {

return false;

}

if ((mes == 4 || mes == 6 || mes == 9 || mes == 11) && dia > 30) {

return false;

}

if (mes == 2) {

bool bisiesto = (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);

if (bisiesto && dia > 29) {

return false;

} else if (!bisiesto && dia > 28) {

return false;

}

}

return true;

}