/\*\*Este Proyecto Incluye Ejercicios Practicos sobre la Recursividad\*\*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

    /\*printNumOfValuesLessThanX();

    printNumberOfAppearances();

    printNumOfEvenValues();

    printSumOfEvenValues();

    getNFromUser();

    printElementOfTheLucasSeries();

    printMaximumValue();

    printMinimumValue();

    printPellNumber();

    printEvenOrOddForSumOfDigits();

    printOddOrEvenForSumOfDigits();

    printDigitAndPositionCorrespondOrNot();

    getTotalNum1AndNum2();

    getTotalAndChar1AndChar2();

    getTotalAndChar1();

    printAscendingOrDescendingOrNeither(); \*/

    printIfNumberOfOccurrencesIsEvenOrOdd();

    return 0;

}

/\*\*Este Procedimiento Calcula la Cantidad de Valores Menores a X\*\*/

#include<stdio.h>

int calculateNumOfValuesLessThanX(int iRange)

{

    int iNum;

    printf("\tIngresa un valor entero (ingresa -1 para terminar el proceso): ");

    scanf("%i", &iNum);

    fflush(stdin);

    if (iNum == -1)

        return 0;

    else if (iNum < iRange)

        return 1 + calculateNumOfValuesLessThanX(iRange);

    else

        return 0 + calculateNumOfValuesLessThanX(iRange);

}

void printNumOfValuesLessThanX()

{

    int iNumber;

    printf("Ingresa X (valor entero): ");

    scanf("%i", &iNumber);

    fflush(stdin);

    printf("El numero de valores menores a X(%i) es %i. ", iNumber, calculateNumOfValuesLessThanX(iNumber));

    printf("\n \n");

}

/\*\*Este Procedimiento Calcula, Dentro de una Secuencia de Caracteres, el Numero de Apariciones de Cierto Caracter\*\*/

#include<stdio.h>

int calculateNumberOfAppearances(char cOriginal)

{

    char cInput;

    printf("\tIngresa un caracter (ingresa '$' para terminar el proceso): ");

    scanf("%c", &cInput);

    fflush(stdin);

    if (cInput == '$')

        return 0;

    else if (cInput == cOriginal)

        return 1 + calculateNumberOfAppearances(cOriginal);

    return calculateNumberOfAppearances(cOriginal);

}

void printNumberOfAppearances()

{

    char cChar;

    printf("Ingresa el caracter a encontrar: ");

    scanf("%c", &cChar);

    fflush(stdin);

    printf("El caracter '%c' fue repetido %i ocasion(es). ", cChar, calculateNumberOfAppearances(cChar));

    printf("\n \n");

}

/\*\*Este Procedimiento Calcula el Numero de Valores Pares de una Secuencia\*\*/

#include<stdio.h>

int calculateNumOfEvenValues()

{

    int iNumber;

    printf("\tIngresa un valor entero (ingresa -1 para terminar el proceso): ");

    scanf("%i", &iNumber);

    fflush(stdin);

    if (iNumber == -1)

        return 0;

    else if (iNumber % 2 == 0)

        return 1 + calculateNumOfEvenValues();

    return calculateNumOfEvenValues();

}

void printNumOfEvenValues()

{

    printf("El numero de valores pares es %i. ", calculateNumOfEvenValues());

    printf("\n \n");

}

/\*\*Este Procedimiento Calcula la Suma de Todos los Valores Pares Ingresados\*\*/

#include<stdio.h>

int calculateSumOfEvenValues()

{

    int iNumber;

    printf("\tIngresa un valor entero (ingresa -1 para terminar el proceso): ");

    scanf("%i", &iNumber);

    fflush(stdin);

    if (iNumber == -1)

        return 0;

    else if (iNumber % 2 == 0)

        return iNumber + calculateSumOfEvenValues();

    return calculateSumOfEvenValues();

}

void printSumOfEvenValues()

{

    printf("La suma de los numeros pares es %i. ", calculateSumOfEvenValues());

    printf("\n \n");

}

/\*\*Este Procedimiento Imprime de 1 a N y Viceversa (N Es Ingresado por el Usuario)\*\*/

#include<stdio.h>

void printFrom1ToN(int iN)

{

    if (iN >= 1)

    {

        printFrom1ToN(iN - 1);

        printf("\t%i\n", iN);

    }

}

void printFromNTo1(int iN)

{

    if (iN >= 1)

    {

        printf("\t%i\n", iN);

        printFromNTo1(iN - 1);

    }

}

void getNFromUser()

{

    int iNumber;

    do{

        printf("Ingresa un valor entero (mayor a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber <= 0)

            printf("ERROR: El numero ingresado debe ser mayor a cero. \n");

    }while (iNumber <= 0);

    printf("Impresion de los Valores 1 a N. \n");

    printFrom1ToN(iNumber);

    printf("Fin de la Impresion de los Valores de 1 a N. \n");

    printf("Impresion de los Valores N a 1. \n");

    printFromNTo1(iNumber);

    printf("Fin de la Impresion de los Valores de N a 1. \n");

    printf("Impresion de los Valores N a 1 a N. \n");

    printFromNTo1ToN(iNumber);

    printf("\nFin de la Impresion de los Valores de N a 1 a N. ");

    printf("\n \n");

}

/\*\*Este Procedimiento Encuentra un Valor de la Serie de Lucas\*\*/

///Serie de Lucas

// F 0 = 2

// F 1 = 1

// F 2 = F 0 + F 1 = 3

// F 3 = F 1 + F 2 = 4

// F 4 = F 2 + F 3 = 7

// F 5 = F 3 + F 4 = 11

// F 6 = F 4 + F 5 = 18

// F 7 = F 5 + F 6 = 29

// F 8 = F 6 + F 7 = 47

#include<stdio.h>

int calculateElementOfTheLucasSeries(int i)

{

    if (i == 0)

        return 2;

    if (i == 1)

        return 1;

    return calculateElementOfTheLucasSeries(i - 1) + calculateElementOfTheLucasSeries(i - 2);

}

void printElementOfTheLucasSeries()

{

    int iNumber;

    do{

        printf("Ingresa un valor entero (mayor o igual a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber < 0)

            printf("ERROR: El numero ingresado debe ser mayor o igual a cero. \n");

    }while (iNumber < 0);

    printf("El elemento #%i de la serie de Lucas es %i. ", iNumber, calculateElementOfTheLucasSeries(iNumber));

    printf("\n \n");

}

/\*\*Este Procedimiento Encuentra el Valor mas Grande de una Secuencia\*\*/

#include<stdio.h>

int calculateMaximumValue(int iRange)

{

    int iNumber, iMaximum;

    if (iRange > 0)

    {

        do{

            printf("\tIngresa un valor (mayor a cero): ");

            scanf("%i", &iNumber);

            fflush(stdin);

            if (iNumber <= 0)

                printf("\tERROR: El numero ingresado debe ser mayor a cero. \n");

        }while (iNumber <= 0);

        iMaximum = calculateMaximumValue(iRange - 1);

        if (iNumber > iMaximum)

            return iNumber;

        else

            return iMaximum;

    }

    return 0;

}

void printMaximumValue()

{

    int iNumber;

    do{

        printf("Ingresa un valor mayor a cero: ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber <= 0)

            printf("ERROR: El numero ingresado debe ser mayor a cero. \n");

    }while (iNumber <= 0);

    printf("El numero mas grande de la secuencia de %i numero(s) es %i. ", iNumber, calculateMaximumValue(iNumber));

    printf("\n \n");

}

/\*\*Este Procedimiento Encuentra el Valor m�s Chico de una Secuencia de Valores\*\*/

#include<stdio.h>

int calculateMinimumValue(int iNum)

{

    int iInputFromUser, iMinimum;

    printf("\tIngresa un valor: ");

    scanf("%i", &iInputFromUser);

    fflush(stdin);

    if (iNum > 1)

    {

        iMinimum = calculateMinimumValue(iNum - 1);

        if (iInputFromUser < iMinimum)

            return iInputFromUser;

        else

            return iMinimum;

    }

    //else

    return iInputFromUser;

}

void printMinimumValue()

{

    int iNumber;

    do{

        printf("Ingresa un valor entero (mayor a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber <= 0)

            printf("ERROR: El numero ingresado debe ser mayor a cero. \n");

    } while (iNumber <= 0);

    printf("El numero mas chico de la secuencia de %i numero(s) es %i. ", iNumber, calculateMinimumValue(iNumber));

    printf("\n \n");

}

/\*\*Este Procedimiento Calcula, Dado un Indice, el Valor de un Numero Pell\*\*/

///Pell Numbers

// F 0 =                 0

// F 1 =                 1

// F 2 = 2 \* F 1 + F 0 = 2

// F 3 = 2 \* F 2 + F 1 = 5

// F 4 = 2 \* F 3 + F 2 = 12

// F 5 = 2 \* F 4 + F 3 = 29

// F 6 = 2 \* F 5 + F 4 = 70

// F 7 =                 169

#include<stdio.h>

int calculatePellNumber(int iN)

{

    if (iN == 0)

        return 0;

    if (iN == 1)

        return 1;

    return 2 \* calculatePellNumber(iN - 1) + calculatePellNumber(iN - 2);

}

void printPellNumber()

{

    int iNumber;

    do{

        printf("Ingresa un valor (mayor o igual a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber < 0)

            printf("ERROR: El numero ingresado debe ser mayor o igual a cero. \n");

    } while (iNumber < 0);

    printf("El numero PELL #%i es %i. ", iNumber, calculatePellNumber(iNumber));

    printf("\n \n");

}

/\*\*Este Procedimiento Determina si la Suma de los Digitos de un Numero es Par o Impar\*\*/

#include<stdio.h>

int calculateEvenOrOddForSumOfDigits(int iNumber)

{

    /\*\*C�digo no Optimizado\*\*/

    /\*int iResultSoFar;

    if (iNumber < 10)

    {

        if(iNumber % 2 == 0)

            return 1;

        else

            return 0;

    }

    iResultSoFar = calculateEvenOrOddForSumOfDigits(iNumber / 10);

    if (iNumber % 10 % 2 == 0)

        if (iResultSoFar == 1)

            return 1;

        else

            return 0;

    else

        if (iResultSoFar == 1)

            return 0;

        else

            return 1;\*/

    /\*\*C�digo Optimizado y Minimizado\*\*/

    if (iNumber < 10)

        return iNumber % 2 == 0 ? 1 : 0;

    if (iNumber % 10 % 2 == 0)

        return calculateEvenOrOddForSumOfDigits(iNumber / 10);

    else

        return !calculateEvenOrOddForSumOfDigits(iNumber / 10);

}

void printEvenOrOddForSumOfDigits()

{

    int iNumber;

    do{

        printf("Ingresa un valor entero (mayor o igual a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber < 0)

            printf("ERROR: El numero ingresado debe ser mayor o igual a cero. \n");

    } while (iNumber < 0);

    printf("La suma de los digitos del numero %i es ", iNumber);

    if (calculateEvenOrOddForSumOfDigits(iNumber) == 1)

        printf("par. ");

    else

        printf("impar. ");

    printf("\n \n");

}

/\*\*Este Procedimiento Determina si la Suma de los Digitos de un Numero es Impar o Par\*\*/

#include<stdio.h>

int calculateOddOrEvenForSumOfDigits(int iNum)

{

    int iResultSoFar;

    if (iNum < 10)

        return iNum % 2 ? 1 : 0;

    iResultSoFar = calculateOddOrEvenForSumOfDigits(iNum / 10);

    if (iResultSoFar == 0)

        if (iNum % 10 % 2 == 0)

            return 0;

        else

            return 1;

    else

        if (iNum % 10 % 2 == 0)

            return 1;

        else

            return 0;

}

void printOddOrEvenForSumOfDigits()

{

    int iNumber;

    do{

        printf("Ingresa un numero entero (mayor o igual a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber < 0)

            printf("ERROR: El numero ingresado debe ser mayor o igual a cero. \n");

    } while (iNumber < 0);

    printf("La suma de digitos del numero ingresado es ");

    if (calculateOddOrEvenForSumOfDigits(iNumber) == 1)

        printf("impar. ");

    else

        printf("par. ");

    printf("\n \n");

}

/\*\*Este Procedimiento Determina si los Digitos de un Numero Corresponden, en Terminos de Par o Impar, a su Respectiva Posicion\*\*/

///Examples

// Example #1

// N = 1234

// Position '0' : 4 --> EVEN & EVEN

// Position '1' : 3 --> ODD & ODD

// Position '2' : 2 --> EVEN & EVEN

// Position '3' : 1 --> ODD & ODD

// RETURN 1 (Todos los Digitos y las Posiciones Corresponden)

//

// Example #2

// N = 56578

// Position '0' : 8 --> EVEN & EVEN

// Position '1' : 7 --> ODD & ODD

// Position '2' : 5 --> EVEN & ODD  <--

// Position '3' : 6 --> ODD & EVEN  <--

// Position '4' : 5 --> EVEN & ODD  <--

// RETURN 0 (No Todos los Digitos y las Posiciones Corresponden)

#include<stdio.h>

int calculateDigitAndPositionCorrespondOrNot(int iDigit, int iPosition)

{

    /// Version ORIGINAL by DAS

    /\*int iAuxiliar;

    if (iDigit < 10)

    {

        //if (iPosition == -1)

        //    iPosition = 0;

        iPosition = iPosition + 1;

        printf("\t\nPosition '%i' : %i", iPosition, iDigit);

        if (iDigit % 2 == iPosition % 2)

            return 1;

        else

            return 0;

    }

    iAuxiliar = calculateDigitAndPositionCorrespondOrNot(iDigit % 10, iPosition);

    if(iAuxiliar == 1)

        return calculateDigitAndPositionCorrespondOrNot(iDigit / 10, iPosition + 1);

    else

        return 0; \*/

    /// Version Optimizada by DAS

    if (iDigit < 10)

    {

        iPosition = iPosition + 1;

        return iDigit % 2 == iPosition % 2 ? 1 : 0;

    }

    if(calculateDigitAndPositionCorrespondOrNot(iDigit % 10, iPosition) == 1)

        return calculateDigitAndPositionCorrespondOrNot(iDigit / 10, iPosition + 1);

    else

        return 0;

}

void printDigitAndPositionCorrespondOrNot()

{

    int iNumber;

    do{

        printf("Ingresa un numero entero (mayor a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber <= 0)

            printf("ERROR: El numero ingresado debe ser mayor a cero. \n");

    } while (iNumber <= 0);

    printf("Los digitos y las posiciones ");

    if (calculateDigitAndPositionCorrespondOrNot(iNumber, -1) == 1)

        printf("si ");

    else

        printf("no ");

    printf("corresponden. ");

    printf("\n \n");

}

/\*\*Este Procedimiento Recibe del Usuario los Siguientes Parametros: TOTAL, NUM1 y NUM2. Despu�s, Procedemos a Imprimir un String Compuesto por los Ultimos Dos Parametros Ingresados\*\*/

#include<stdio.h>

void printSequenceOfTotalNum1AndTotalNum2(int iTotal, int iNum1, int iNum2)

{

    if (iTotal > 0)

    {

        printf("%i", iNum1);

        printSequenceOfTotalNum1AndTotalNum2(iTotal - 1, iNum1, iNum2);

        printf("%i", iNum2);

    }

}

void getTotalNum1AndNum2 ()

{

    int iTotal, iNum1, iNum2;

    do {

        printf("Ingresa un numero entero (mayor a cero): ");

        scanf("%i", &iTotal);

        fflush(stdin);

        if (iTotal <= 0)

            printf("El numero ingresado debe ser mayor a cero. \n");

    } while (iTotal <= 0);

    printf("\tIngresa un numero entero: ");

    scanf("%i", &iNum1);

    fflush(stdin);

    printf("\tIngresa otro numero entero: ");

    scanf("%i", &iNum2);

    fflush(stdin);

    printf("Impresion de la secuencia de %i '%i's y %i '%i's: \n\t", iTotal, iNum1, iTotal, iNum2);

    printSequenceOfTotalNum1AndTotalNum2(iTotal, iNum1, iNum2);

    printf("\n \n");

}

/\*\*Este Procedimiento Lee del Usuario lo Siguiente: 1 Numero(N) y 2 Caracteres. Despues, la Rutina Procede a Imprimir el Primer Caracter N Veces e Imprime el Segundo Caracter 2 x N Veces\*\*/

#include<stdio.h>

void printSequenceOfTotalChar1And2TotalChar2(int iTotal, char cChar1, char cChar2)

{

    if (iTotal > 0)

    {

        printf("%c", cChar1);

        printSequenceOfTotalChar1And2TotalChar2(iTotal - 1, cChar1, cChar2);

        printf("%c%c", cChar2, cChar2);

    }

}

void getTotalAndChar1AndChar2 ()

{

    int iTotal;

    char cChar1, cChar2;

    do{

        printf("Ingresa un valor (mayor a cero): ");

        scanf("%i", &iTotal);

        fflush(stdin);

        if (iTotal <= 0)

            printf("ERROR: El valor ingresado debe ser mayor a cero. \n");

    } while (iTotal <= 0);

    printf("\tIngresa un caracter: ");

    scanf("%c", &cChar1);

    fflush(stdin);

    printf("\tIngresa otro caracter: ");

    scanf("%c", &cChar2);

    fflush(stdin);

    printf("Impresion de la secuencia de %i '%c's y %i '%c's: \n\t", iTotal, cChar1, iTotal \* 2, cChar2);

    printSequenceOfTotalChar1And2TotalChar2(iTotal, cChar1, cChar2);

    printf("\n \n");

}

/\*\*Este Procedimiento se Encarga de Leer del Usuario un Numero(N) y una Letra Minuscula. Despu�s, la Rutina Procede a Imprimir N Letras Minusculas y N Letras Mayusculas\*\*/

#include<stdio.h>

int printSequenceOfTotalChar1AndTotalUppercaseChar1(int iTotal, char cChar1)

{

    if (iTotal > 0)

    {

        printf("%c", cChar1);

        printSequenceOfTotalChar1AndTotalUppercaseChar1(iTotal - 1, cChar1);

        printf("%c", cChar1 - 32);

    }

}

void getTotalAndChar1()

{

    int iTotal;

    char cChar1;

    do{

        printf("Ingresa un valor entero (mayor a cero): ");

        scanf("%i", &iTotal);

        fflush(stdin);

        if (iTotal <= 0)

            printf("ERROR: El numero ingresado debe ser mayor a cero. \n");

    }while (iTotal <= 0);

    do{

        printf("Ingresa un caracter (a-z): ");

        scanf("%c", &cChar1);

        fflush(stdin);

        if (!(cChar1 >= 97 && cChar1 <= 122))

            printf("ERROR: El caracter ingresado debe ser entre a y z. \n");

    }while (!(cChar1 >= 97 && cChar1 <= 122));

    printf("Impresion de %i '%c's y %i '%c's: \n\t", iTotal, cChar1, iTotal, cChar1 - 32);

    printSequenceOfTotalChar1AndTotalUppercaseChar1(iTotal, cChar1);

    printf("\n \n");

}

/\*\*Este Procedimiento Se Encarga Determinar si los Digitos de un Numero son Ascendentes, Descendentes o Ninguno de los Dos\*\*/

///Instrucciones

// La funcion lee un numero de izquierda a derecha y regresa:

//      1, si los digitos son ascendentes

//     -1, si los digitos son descendentes

//      0, si los digitos no se comportan de manera ascendente ni de manera descendente.

///Asunciones

// El numero que recibe la funcion siempre tiene 2 o mas digitos

// El numero que recibe la funcion no tiene digitos repetidos

#include<stdio.h>

int calculateAscendingOrDescendingOrNeither(int iNumber)

{

    /// Version Original No Optimizada by DAS

    /\*int iResultSoFar;

    if (iNumber < 100)

    {

        if (iNumber % 10 > iNumber / 10)

            return 1;

        else

            return -1;

    }

    iResultSoFar = calculateAscendingOrDescendingOrNeither(iNumber % 100);

    if (iResultSoFar == 1)

    {

        if (calculateAscendingOrDescendingOrNeither(iNumber / 10) == 1)

            return 1;

        else if(calculateAscendingOrDescendingOrNeither(iNumber / 10) == -1)

            return 0;

    }

    else if (iResultSoFar == -1)

    {

        if (calculateAscendingOrDescendingOrNeither(iNumber / 10) == - 1)

            return - 1;

        else if (calculateAscendingOrDescendingOrNeither(iNumber / 10) == 1)

            return 0;

    }\*/

    ///Version Optimizada by DAS

    if (iNumber < 100)

        return iNumber % 10 > iNumber / 10 ? 1 : -1;

    if (calculateAscendingOrDescendingOrNeither(iNumber % 100) == 1)

        return calculateAscendingOrDescendingOrNeither(iNumber / 10) == 1 ? 1 : 0;

    else if (calculateAscendingOrDescendingOrNeither(iNumber % 100) == -1)

        return calculateAscendingOrDescendingOrNeither(iNumber / 10) == - 1 ? -1 : 0;

}

void printAscendingOrDescendingOrNeither()

{

    int iNumber;

    do{

        printf("Ingresa un numero (mayor o igual a diez): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber < 10)

            printf("ERROR: El numero ingresado debe ser mayor o igual a diez. \n");

    }while (iNumber < 10);

    printf("La secuencia de digitos del numero ingresado (de izquierda a derecha) es ");

    if (calculateAscendingOrDescendingOrNeither(iNumber) == 1)

        printf("ascendente. ");

    else if (calculateAscendingOrDescendingOrNeither(iNumber) == -1)

        printf("descendente. ");

    else

        printf("ni ascendente ni descendente. ");

    printf("\n \n");

}

/\*\*Este Procedimiento Determina Cuantas Veces(N) Aparece un Digito (0-9) en un Numero. La Rutina Procede a Determinar si la Cantidad de Apariciones es Par o Impar\*\*/

///Instrucciones

// La funcion regresa 1 si la cantidad de apariciones es par

// La funcion regresa 0 si la cantidad de apariciones es impar

///Ejemplos

// num = 124, digit = 2 --> RETURN 0 (1 aparicion - impar)

// num = 12342, digit = 2 --> RETURN 1 (2 apariciones - par)

// num = 10200240, digit = 0 --> RETURN 1 (4 apariciones - par)

#include<stdio.h>

int calculateIfNumberOfOccurrencesIsEvenOrOdd(int iNumber, int iDigit)

{

    ///Version Original No Optimizada by DAS

    /\*if (iNumber < 10)

    {

        if (iNumber == iDigit)

            return 0;

        else return 1;

    }

    if (iNumber % 10 != iDigit)

    {

        if (calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber / 10, iDigit) == 1)

            return 1;

        else if (calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber / 10, iDigit) == 0)

            return 0;

    }

    else if (iNumber % 10 == iDigit)

    {

        if (calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber / 10, iDigit) == 0)

            return 1;

        else if (calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber / 10, iDigit) == 1)

            return 0;

    }\*/

    ///Version Optimizada by DAS

    if (iNumber < 10)

        return iNumber == iDigit ? 0 : 1;

    if (iNumber % 10 != iDigit)

        return calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber / 10, iDigit);

    else

        return !calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber / 10, iDigit);

}

void printIfNumberOfOccurrencesIsEvenOrOdd()

{

    int iNumber, iDigit;

    do{

        printf("Ingresa un numero (mayor o igual a cero): ");

        scanf("%i", &iNumber);

        fflush(stdin);

        if (iNumber < 0)

            printf("ERROR: El numero ingresado debe ser mayor o igual a cero. \n");

    }while (iNumber < 0);

    do{

        printf("Ingresa un digito (0-9): ");

        scanf("%i", &iDigit);

        fflush (stdin);

        if (!(iDigit >= 0 && iDigit <= 9))

            printf("ERROR: El numero ingresado debe ser un digito entre 0 y 9. \n");

    }while (!(iDigit >= 0 && iDigit <= 9));

    printf("El numero de apariciones del digito %i en el numero %i es ", iDigit, iNumber);

    if (calculateIfNumberOfOccurrencesIsEvenOrOdd(iNumber, iDigit) == 1)

        printf("par. ");

    else

        printf("impar. ");

    printf("\n \n");

}

/\*\*Este Procedimiento de N a 1 a N\*\*/

#include<stdio.h>

void printFromNTo1ToN(iN)

{

    if (iN >= 1)

    {

        printf("%i ", iN);

        printFromNTo1ToN(iN - 1);

        if (iN != 1)

            printf("%i ", iN);

    }

}