

```

/**Este Proyecto Trata sobre Ejercicios Prácticos de las Funciones**/

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

int main()
{
    int iOption;
    char cContinua[10];
    do{
        printf("1.  Calcular el Area de un Rectangulo\n");
        printf("2.  Encontrar el Mayor Dígito de un Numero de 2 Dígitos\n");
        printf("3.  Calcular el Factorial de un Numero\n");
        printf("4.  Determinar si un Entero es Par o Impar\n");
        printf("5.  Imprimir un Entero del Largo Deseado\n");
        printf("6.  Revertir una Letra Minuscula a Mayuscula y
Viceversa\n");
        printf("7.  Concatenacion de 3 Enteros Positivos entre 0 y 9\n");
        printf("8.  Encontrar Todos los Dígitos de un Numero que Sean
Menores a Cierta Entero\n");
        printf("9.  Calcular la Suma de los Divisores de un Numero\n");
        printf("10. Determinar si un Numero es Perfecto o NO\n\n");
        printf("Escoge una opcion del Menu: ");
        scanf("%i", &iOption);
        fflush(stdin);

        printf("\n-----
-----\n\n");

        switch (iOption)
        {
            case 1:
                //Calcular el Area de un Rectangulo
                printArea();
                break;
            case 2:
                //Encontrar el Mayor Dígito de un Numero de 2 Dígitos
                printLargerDigit();
                break;
            case 3:
                //Calcular el Factorial de un Numero
                printFactorial();
                break;
            case 4:

```

```

        //Determinar si un Entero es Par o Impar
        printEvenOrOdd();
        break;
    case 5:
        //Imprimir un Entero del Largo Deseado
        printDesiredNumber();
        break;
    case 6:
        //Revertir una Letra Minuscula a Mayuscula y Viceversa
        printReverseLetter();
        break;
    case 7:
        //Concatenacion de 3 Enteros Positivos entre 0 y 9
        printSumOfCharacters();
        break;
    case 8:
        //Encontrar Todos los Digos de un Numero que Sean Menores a
Cierta Entero
        printLeastDigitsThan();
        break;
    case 9:
        //Calcular la Suma de los Divisores de un Numero
        printSumOfDivisors();
        break;
    case 10:
        //Determinar si un Numero es Perfecto o NO
        printPerfectOrNot();
        break;
    default:
        printf("ERROR: Has seleccionado una opcion inexistente. \n");
        break;
}

printf("\nDeseas regresar al MENU (si/no): ");
scanf("%s", &cContinua);

printf("\n");

} while (strcmp(cContinua, "Si") == 0 || strcmp(cContinua, "si") == 0 ||
strcmp(cContinua, "SI") == 0);

printf("Hasta luego. Regresa pronto!\n\n");

return 0;

```

```

}

//Calcular el Area de un Rectangulo

float calculateArea(float fH, float fB)
{
    float fArea;
    fArea = fH * fB;
    return fArea;
}

void printArea()
{
    float fHeight, fBase;
    printf("Ingresa la altura del rectangulo: ");
    scanf("%f", &fHeight);
    printf("Ingresa la base del rectangulo: ");
    scanf("%f", &fBase);

    printf("El area del rectangulo es: %.2f", calculateArea(fHeight,
fBase));

    printf("\n\n");
}

//Encontrar el Mayor Dígito de un Número de 2 Dígitos

int calculateDigit(int iNumber)
{
    if (iNumber > 9 && iNumber <100)
    {
        if (iNumber % 10 > iNumber / 10)
            return iNumber % 10;
        else
            return iNumber / 10;
    }
    else
    {
        printf("ERROR: El valor ingresado no cuenta con 2 digitos. \n");
        return -1;
    }
}

void printLargerDigit()
{

```

```

    int iTwoDigits, iMaxDigit;

    do{
        printf("Ingresa un numero entero (positivo) de 2 digitos (mayores a
cero): ");
        scanf("%i", &iTwoDigits);

        iMaxDigit = calculateDigit(iTwoDigits);
    } while (iMaxDigit == -1);

    printf("El mayor digito del numero %i es %i", iTwoDigits, iMaxDigit);

    printf("\n \n");
}

//Calcular el Factorial de un Numero

int calculateFactorial (int iNumber)
{
    int iFactorial = 1, f;
    if (iNumber >= 0)
    {
        for (f = iNumber; f > 0; f--)
        {
            iFactorial = iFactorial * f;
        }
        return iFactorial;
    }
    else
    {
        printf("ERROR: El valor ingresado es negativo.\n");
        return -1;
    }
}

void printFactorial ()
{
    int iNumber, iFactorial;

    do{
        printf ("Ingresa un numero entero y positivo: ");
        scanf("%i", &iNumber);

        iFactorial = calculateFactorial(iNumber);
    }
}

```

```

    }while (iFactorial < 0);

    printf("El factorial del %i es %i. ", iNumber, iFactorial);

    printf("\n\n");
}

//Determinar si un Entero es Par o Impar

int calculateEvenOrOdd(int iNumber)
{
    if (iNumber % 2 == 0)
        return 1;
    else
        return 0;
}

void printEvenOrOdd()
{
    int iValue, iType;
    printf("Ingresa un valor entero: ");
    scanf("%i", &iValue);

    iType = calculateEvenOrOdd(iValue);

    if (iType == 1)
        printf("El valor %i es par. ", iValue);
    else
        printf("El valor %i es impar. ", iValue);

    printf("\n\n");
}

//Imprimir un Entero del Largo Deseado

unsigned long long calculateDesiredNumber(int iLength)
{
    int f;
    unsigned long long iDesiredNumber = 0, iTen = 1;
    if (iLength <= 0)
    {
        printf("ERROR: el largo del entero debe ser mayor a cero. \n");
        return -1;
    }
}

```

```

else
{
    if (iLength < 10)
    {
        for (f = iLength; f > 0; f--)
        {
            iDesiredNumber = iDesiredNumber + (f * iTen);
            iTen = iTen * 10;
        }
    }
    else
    {
        for (f = 1; f <= iLength; f++)
        {
            iDesiredNumber = iDesiredNumber + (9 * iTen);
            iTen = iTen * 10;
        }
    }
    return iDesiredNumber;
}
}

void printDesiredNumber()
{
    int iLength;
    unsigned long long iDesiredNumber;

    do{
        printf("Ingresa el largo del entero: ");
        scanf("%i", &iLength);

        iDesiredNumber = calculateDesiredNumber(iLength);
    }while (iDesiredNumber == -1);

    printf("El valor entero (con un largo de %i) generado es %llu. ",
iLength, iDesiredNumber);

    printf("\n\n");
}

//Revertir una Letra Minuscula a Mayuscula y Viceversa

char calculateReverseLetter(char cLetra)
{

```

```

    if (cLetra >= 65 && cLetra <= 90)
        return cLetra + 32;
    else if (cLetra >= 97 && cLetra <= 122)
        return cLetra - 32;
    else
    {
        printf("ERROR: El caracter no es una letra minuscula ni mayuscula.
\n");
        return '0';
    }
}

void printReverseLetter()
{
    char cLetra, cLetraReversa;

    do{
        fflush(stdin);
        printf("Ingresa una letra (mayuscula o minuscula): ");
        scanf("%c", &cLetra);
        cLetraReversa = calculateReverseLetter(cLetra);
    }while (cLetraReversa == '0');

    printf("La letra revertida de %c es %c. ", cLetra, cLetraReversa);

    printf("\n\n");
}

//Concatenacion de 3 Enteros Positivos entre 0 y 9

int calculateSumOfCharacters(char cInt1, char cInt2, char cInt3)
{
    if ((cInt1 >= 48 && cInt1 <= 57) && (cInt2 >= 48 && cInt2 <= 57) &&
(cInt3 >= 48 && cInt3 <= 57))
    {
        return ( (cInt1 - 48) * 100 + (cInt2 - 48) * 10 + (cInt3 - 48) );
    }
    else
    {
        printf("ERROR: Los digitos ingresados no se encuentran entre 0 y 9.
\n");
        return -1;
    }
}

```

```

void printSumOfCharacters()
{
    char cInteger1, cInteger2, cInteger3;
    int iSum;

    do{
        printf("Ingresa un digito (0-9): ");
        scanf("%c", &cInteger1);
        fflush(stdin);
        printf("Ingresa un digito (0-9): ");
        scanf("%c", &cInteger2);
        fflush(stdin);
        printf("Ingresa un digito (0-9): ");
        scanf("%c", &cInteger3);
        fflush(stdin);

        iSum = calculateSumOfCharacters(cInteger1, cInteger2, cInteger3);
    }while (iSum == -1);

    printf("La concatenacion de los digitos %c, %c y %c es %003i. ",
cInteger1, cInteger2, cInteger3, iSum);

    printf("\n\n");
}

//Encontrar Todos los Digitos de un Numero que Sean Menores a Cierta Entero

float calculateAverage(int iNumber, int iLimit)
{
    int iCounter = 0, iSum = 0;
    //int iTen = 10, iValidator = 0, iAuxiliar, iDoubleNumber = 0;
    //float fAverage;
    if (!(iNumber > 0 && iLimit > 0))
    {
        printf("Error: Los numeros ingresados deben ser mayores a cero.
\n");
        return -1;
    }
    else
    {
        printf("NUMERO = %i\tLIMITE = %i\nLos digitos del NUMERO que son
menores al LIMITE: \n", iNumber, iLimit);
        do{

            if (iNumber % 10 < iLimit)

```



```

        {
            printf("- %i\n", iNumber % 10);
            iCounter++;
            iSum += iNumber % 10;
        }

        iNumber = iNumber / 10;

        /*Procedimiento no Optimizado*/
        /*
        iAuxiliar = iDoubleNumber - (iDoubleNumber / iTen * iTen);
        iDoubleNumber = iDoubleNumber - iAuxiliar;
        iValidator = iValidator + iAuxiliar;
        iAuxiliar = iAuxiliar / (iTen / 10);
        if (iAuxiliar < iLimit)
        {
            printf("- %i\n", iAuxiliar);
            iSum = iSum + iAuxiliar;
            iCounter++;
        }
        iTen = iTen * 10; */

    } while (iNumber > 0);

    //fAverage = ((float) iSum) / iCounter;

    return ((float) iSum) / iCounter;
}
}

void printLeastDigitsThan()
{
    int iNumber, iLimit;
    float fAverage;

    do{
        printf("Ingresa un numero entero mayor a cero: ");
        scanf("%i", &iNumber);
        fflush(stdin);
        printf("Ingresa otro numero entero mayor a cero (este numero sera
usado para comparar): ");
        scanf("%i", &iLimit);
        fflush(stdin);

        fAverage = calculateAverage(iNumber, iLimit);
    } while (iNumber > iLimit);
}

```

```

    } while (fAverage == -1);

    printf("El promedio de los numeros es %.2f", fAverage);

    printf("\n \n");
}

//Calcular la Suma de los Divisores de un Numero

int calculateSumOfDivisors(int iLimit, int lPrintDivisors)
{
    /*Procedimiento no Optimizado*/
    /*int f, iSum = 0;

    printf("Suma de los Divisores de %i: \n", iLimit);

    for (f = 1; f <= iLimit; f++)
    {
        if(iLimit % f == 0)
        {
            iSum += f;
            printf("\t+ %i\n", f);
        }
    }

    return iSum; */

    /*Procedimiento Optimizado*/
    int f, iSum = 0;

    if (iLimit <= 0)
    {
        printf("ERROR: El valor ingresado debe ser mayor a cero. \n");
        return -1;
    }

    if (lPrintDivisors == 1)
        printf("La Suma de los Divisores de %i: \n", iLimit);

    for (f = 1; f * f <= iLimit; f++)
    {
        if (iLimit % f == 0)
        {

```

```

        if (!(f * f == iLimit))
        {
            if (lPrintDivisors == 1)
                printf("\t\t+ %i\n\t+ %i\n", f, iLimit / f);
            iSum += f + (iLimit / f);
        }
        else
        {
            if (lPrintDivisors == 1)
                printf("\t\t+ %i\n", f);
            iSum += f;
        }
    }
}

if(lPrintDivisors == 1)
    printf("-----\n");

return iSum;
}

```

```

void printSumOfDivisors()
{
    int iNumber, iSum;
    do{
        printf("Ingresa un numero entero mayor a 0: ");
        scanf("%i", &iNumber);

        iSum = calculateSumOfDivisors(iNumber, 1);
    } while (iSum == -1);

    printf("\t= %i", iSum);

    printf("\n\n");
}

```

//Determinar si un Numero es Perfecto o NO

```

void printPerfectOrNot()
{
    int iNumber, iSum;

    do{
        printf("Ingresa un valor entero mayor que cero: ");
        scanf("%i", &iNumber);
    }
}

```

```
        iSum = calculateSumOfDivisors(iNumber, 0);

    }while (iSum == -1);

    if (iSum - iNumber == iNumber)
        printf("El numero ingresado, %i, es un numero perfecto, ya que la
suma de sus divisores (sin incluir a si mismo) es igual a este mismo. ",
iNumber);
    else
        printf("El numero ingresado, %i, no es un numero perfecto, ya que la
suma de sus divisores (sin incluir a si mismo) no es igual a este mismo. ",
iNumber);

    printf("\n\n");
}
```