

**MATERIA**

Algoritmos

**NOMBRE DEL ESTUDIANTE**

Matías Llumiquinga

Emilia Cano

**TAREA**

Semana 16: Proyecto Integrador

**FECHA**

08-07-2024

**ISWZ1101- ALGORITMOS PROYECTO INTEGRADOR**

**OBJETIVO PROPUESTO DE LA CONSIGNA:**

El trabajo por realizar tiene como finalidad aplicar el conocimiento adquirido en el análisis y solución de problemas computacionales y creación de un algoritmo que dé solución al problema planteado.

**INDICACIONES:**

Los estudiantes deben analizar el problema planteado por el docente con el cual se debe: Definir claramente los datos y procesos involucrados en el problema, proponer y evaluar la mejor alternativa para resolver al problema, finalmente seleccionar la mejor alternativa de solución fundamentado en los principios de la algoritmia. Se debe presentar el algoritmo a través de un diagrama de flujo, pseudocódigo las pruebas de escritorio respectivas. Como parte final el estudiante debe realizar la implementación de la solución en un lenguaje de programación estructurado preferiblemente C. Los problemas propuestos se encuentran como anexos.

**Solución:**

Analizar el problema seleccionado, identifique claramente los datos de entrada, procesos y salida. En función de los conocimientos adquiridos plantee alternativas de diseño de la solución usando Diagrama de Flujo para cada una de ellas. Recuerde que el diagrama debe tener identificado los datos y procesos. Luego debe crear el pseudocódigo del algoritmo optimizado. Y por último se debe hacer la implementación del algoritmo en lenguaje de programación C.

**FORMA DE TRABAJO:**

La propuesta se la desarrollará en parejas.

**ESPECIFICACIONES DE ENTREGA:**

**Informe, el** estudiante debe realizar lo siguiente:

* Formulación del problema identificando sus principales variables: Datos de entrada, salida, procesos involucrados.
* Alternativas de solución al problema: Plantear por lo menos 2 soluciones algorítmica y seleccionar la mejor alternativa, para cada alternativa se debe presentar el diagrama de flujo.
* Analiza las alternativas que dan solución al problema: Se debe justificar con argumentos técnicos las soluciones planteadas.
* Fundamentación de la solución al problema. Se debe seleccionar la mejor solución al problema planteado, argumentando su selección. De la solución elegida presentar pseudocódigo y pruebas de escritorio.
* Se debe presentar un documento de los puntos anteriores, además se debe presentar el programa en lenguaje C de la solución al problema, y debe subirse al GIT y compartir el enlace.

**Presentación**de su proyecto con ambas soluciones, análisis comparativo de las mismas incluyendo conclusiones,**máximo 8 diapositivas**, utilizando el formato de presentación adjunta.

**Archivo fuente,** de las soluciones de su proyecto

**Colocar el Informe, Presentación y Archivos Fuente en un comprimido .zip y subirlo cada miembro del grupo**

# PROBLEMA SELECCIONADO:

**Opción 10: Multiplicación rusa**

Construir un algoritmo que permita multiplicar dos números enteros positivos empleando el método denominado MULTIPLICACIÓN RUSA. Este método permite calcular el producto de M\*N de la siguiente manera:

En pasos sucesivos se divide M por 2 (división entera) y se multiplica N por 2. Este proceso se repite hasta que M es 0. El resultado de la multiplicación deseada se obtiene acumulando aquellos valores sucesivos de N para los cuales el valor de M es impar:

Ejemplo 1: 31 \* 27

N M Acumulado

31\* 27 31

62\* 13 31+62

124 6 31+62

248\* 3 31+62+248

496\* 1 31+62+248+496

992 0

Ejemplo 2: 25 \* 6

N M Acumulado

25 6 0

50\* 3 0+50

100\* 1 0+50+100

200 0

# PRIMERA SOLUCIÓN:

## 1) Análisis Del Problema:



## 2) Pseudocódigo:

Algoritmo Multiplicacion\_Rusa

Definir m, n, mProcess, nProcess, acum Como Entero;

acum = 0;

Repetir

Mostrar "Ingrese el multiplicando (M) (Entre 0-10000):";

Leer m;

Mostrar "Ingrese el multiplicador (N) (Entre 0-10000):";

Leer n;

Si (m < 0) o (n < 0) o (m > 10000) o (n > 10000) Entonces

Mostrar "Al menos un número ingresado está fuera del rango definido o tiene decimales";

Mostrar "Ingrese números enteros positivos dentro del rango";

Fin Si

Hasta Que (m >= 0) y (n >= 0) y (m <= 10000) y (n <= 10000)

mProcess = m;

nProcess = n;

Repetir

Mostrar "M: ", mProcess;

Si ((mProcess MOD 2) == 1) Entonces

acum = acum + nProcess;

Mostrar "N: ", nProcess, "\*";

SiNo

Mostrar "N: ", nProcess;

Fin Si

Mostrar "Acumulado: ", acum;

mProcess = trunc(mProcess/2);

nProcess = nProcess\*2;

Si (mProcess==0) y (m<>0) Entonces

Mostrar "M: ", mProcess;

Mostrar "N: ", nProcess;

Mostrar "Acumulado: ", acum;

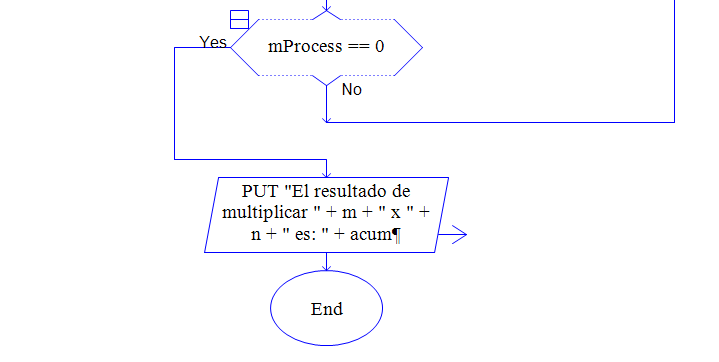
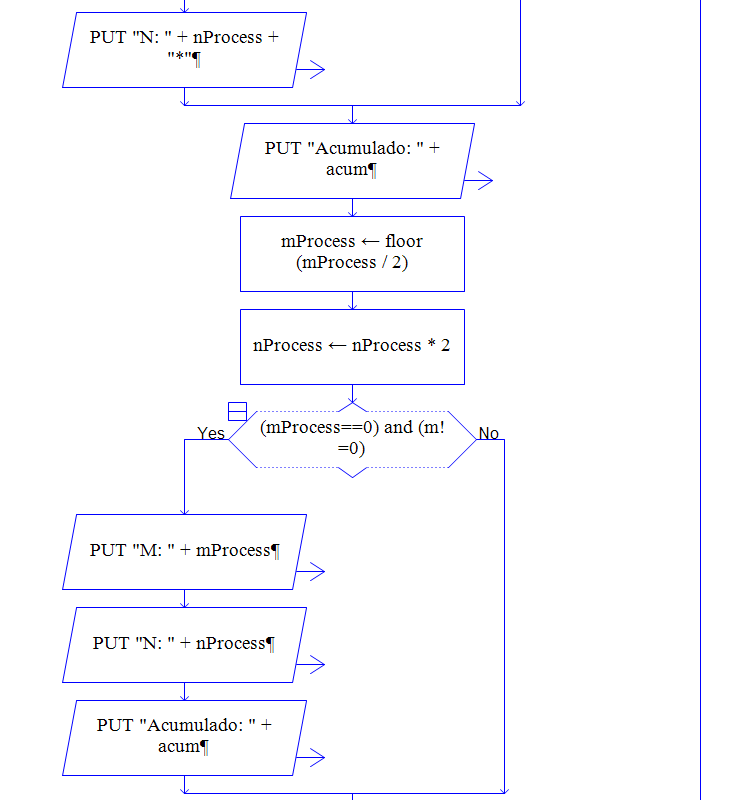
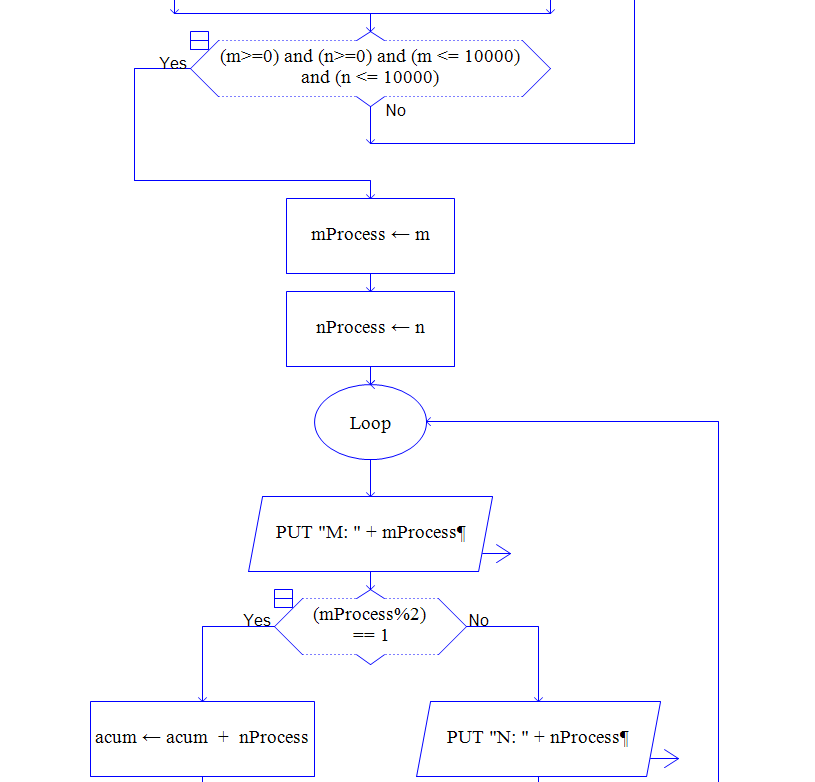
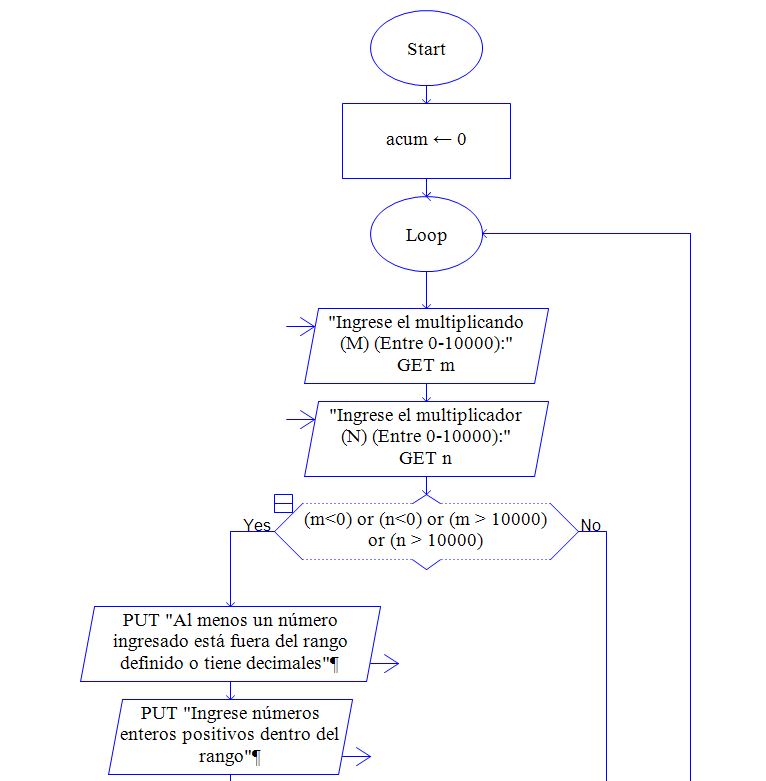
Fin Si

Hasta Que (mProcess == 0)

Mostrar "El resultado de multiplicar ", m " x ", n, " es: ", acum;

FinAlgoritmo

## 3) Diagrama de Flujo:



## 4) Código en C:

//#include <limits.h>    //printf("%i\n", INT\_MAX);

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <conio.h>

#include <time.h>

int main(int *argc*, char const \**argv*[])

{

    setlocale(LC\_CTYPE, "es\_ES.UTF-8");

    int m, n, mProcess, nProcess, acum;

*clock\_t* start, end;

    double cpu\_time\_used;

    acum = 0;

    m = 0;

    n = 0;

    mProcess = 0;

    nProcess = 0;

    do

    {

        printf("Ingrese el multiplicando (M) (Entre 0-10000):\n");

        scanf("%i", &m);

        printf("Ingrese el multiplicador (N) (Entre 0-10000):\n");

        scanf("%i", &n);

        if (m<0 || n<0 || m>10000 || n>10000)

        {

            printf("Al menos un número ingresado está fuera del rango definido o tiene decimales\n");

            printf("Ingrese números enteros positivos dentro del rango\n");

        }

    } while (m<0 || n<0 || m>10000 || n>10000);

    //--------------------------------------START TIME--------------------------------------

    start = clock();

    //--------------------------------------END TIME--------------------------------------

    mProcess = m;

    nProcess = n;

    do

    {

        printf("M: %i\n", mProcess);

        if ((mProcess%2)==1)

        {

            acum = acum + nProcess;

            printf("N: %i\*\n", nProcess);

        }

        else

        {

            printf("N: %i\n", nProcess);

        }

        printf("Acumulado: %i\n", acum);

        mProcess = trunc(mProcess/2);

        nProcess = nProcess\*2;

        if (mProcess==0 && m!=0)

        {

            printf("M: %i\n", mProcess);        //Permite observar el caso en el que mProcess llega a 0

            printf("N: %i\n", nProcess);

            printf("Acumulado: %i\n", acum);

        }

    } while (mProcess > 0);

    printf("El resultado de multiplicar %i x %i, es: %i\n", m,n,acum);

    //--------------------------------------START TIME--------------------------------------

    end = clock();

    cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;

    printf("Tiempo de ejecución: %f\n", cpu\_time\_used);

    //--------------------------------------END TIME--------------------------------------

    getch();

    return 0;

}

## 5) Pruebas de Escritorio:





# SEGUNDA SOLUCIÓN:

## 1) Análisis Del Problema:



## 2) Pseudocódigo:

Algoritmo Multiplicacion\_Rusa

Definir multiplicandoM, multiplicadorN, procesoM, procesoN, acumulador,cantidadDiv, i Como Entero;

acumulador = 0;

cantidadDiv = 0;

Mostrar "Ingrese el multiplicando (M) (Entre 0-10000):";

Leer multiplicandoM;

Mostrar "Ingrese el multiplicador (N) (Entre 0-10000):";

Leer multiplicadorN;

Mientras (multiplicandoM < 0) o (multiplicadorN < 0) o (multiplicandoM > 10000) o (multiplicadorN > 10000) Hacer

Mostrar "Al menos un número ingresado está fuera del rango definido o tiene decimales";

Mostrar "Ingrese números enteros positivos dentro del rango";

Mostrar "Ingrese el multiplicando (M) (Entre 0-10000):";

Leer multiplicandoM;

Mostrar "Ingrese el multiplicador (N) (Entre 0-10000):";

Leer multiplicadorN;

Fin Mientras

procesoM = multiplicandoM;

procesoN = multiplicadorN;

Mientras (procesoM > 0) Hacer

cantidadDiv = cantidadDiv + 1;

procesoM = trunc(procesoM/2);

Fin Mientras

procesoM = multiplicandoM;

Para i=0 Hasta cantidadDiv Con Paso 1 Hacer

Mostrar "M: ", procesoM;

Si ((procesoM MOD 2) == 1) Entonces

acumulador = acumulador + procesoN;

Mostrar "N: ", procesoN, "\*";

SiNo

Mostrar "N: ", procesoN;

Fin Si

Mostrar "Acumulado: ", acumulador;

procesoM = trunc(procesoM/2);

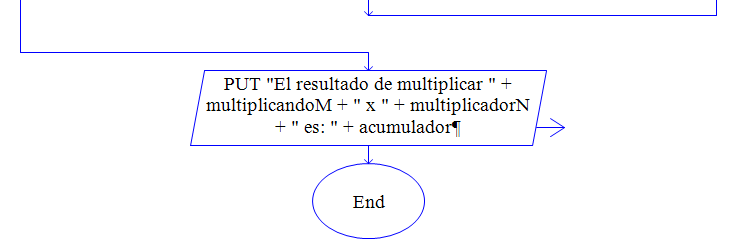
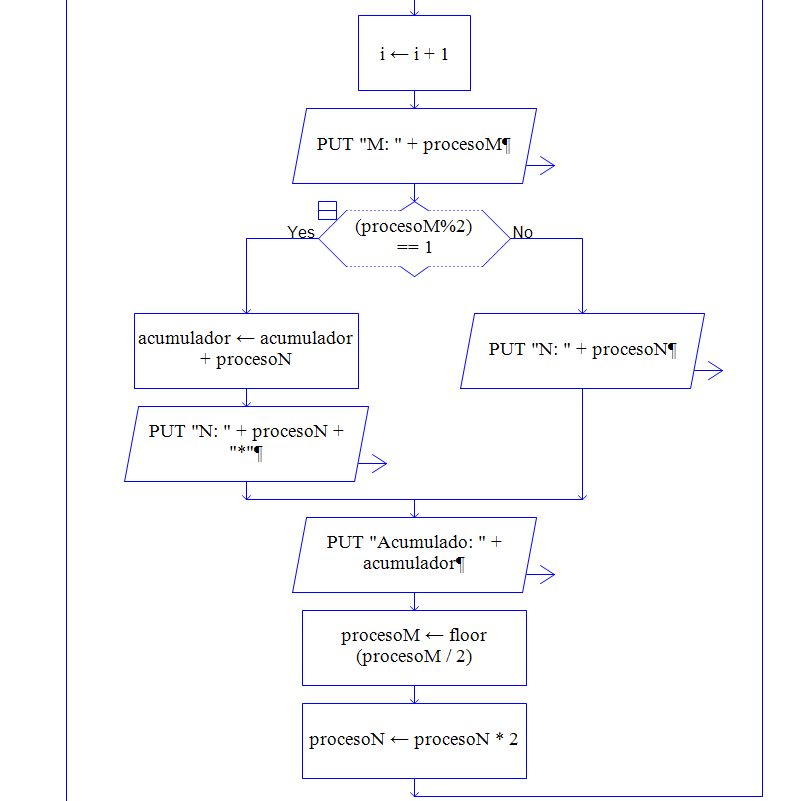
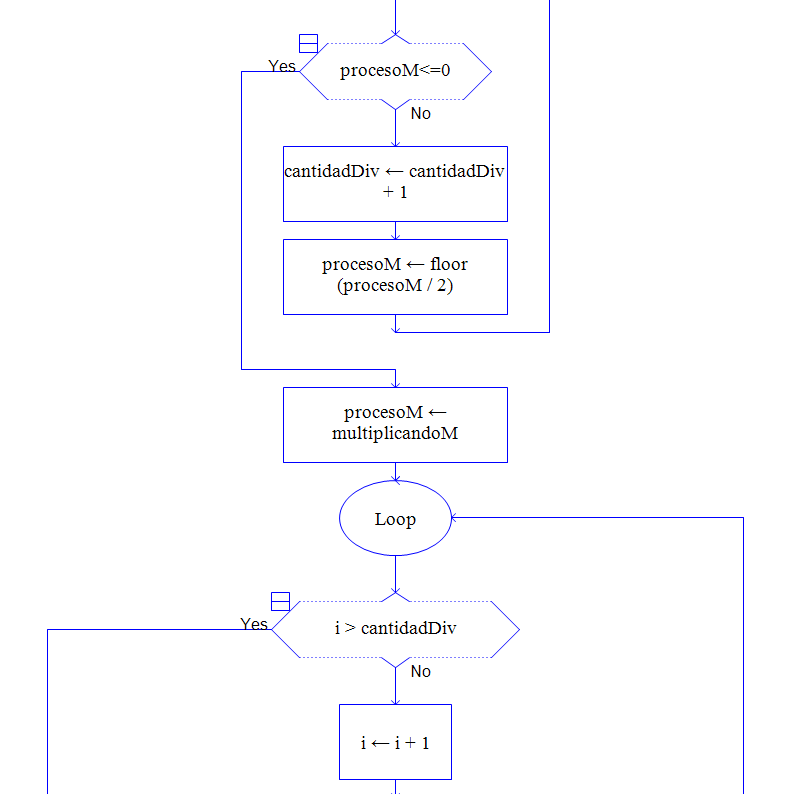
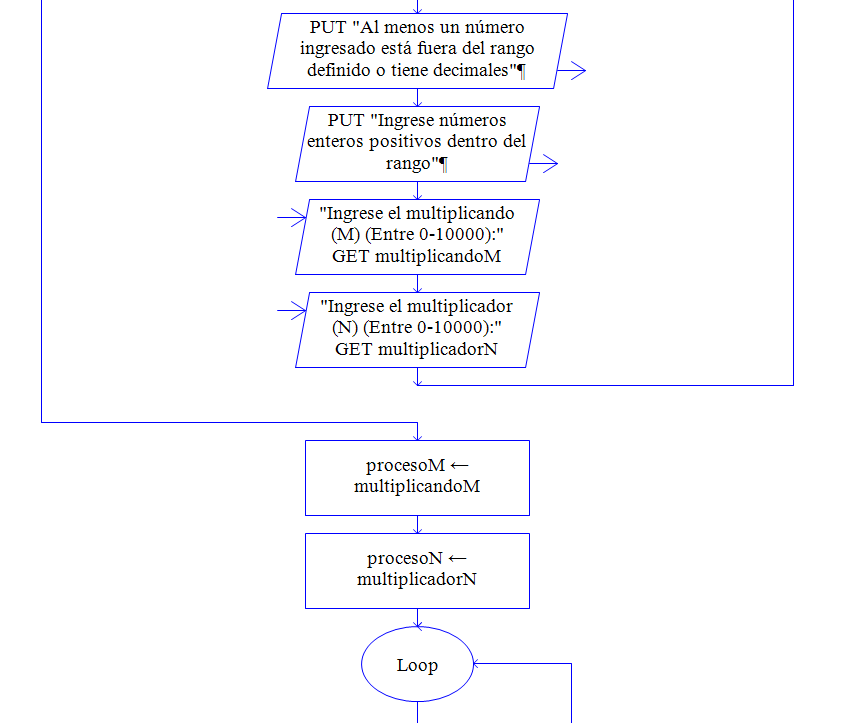
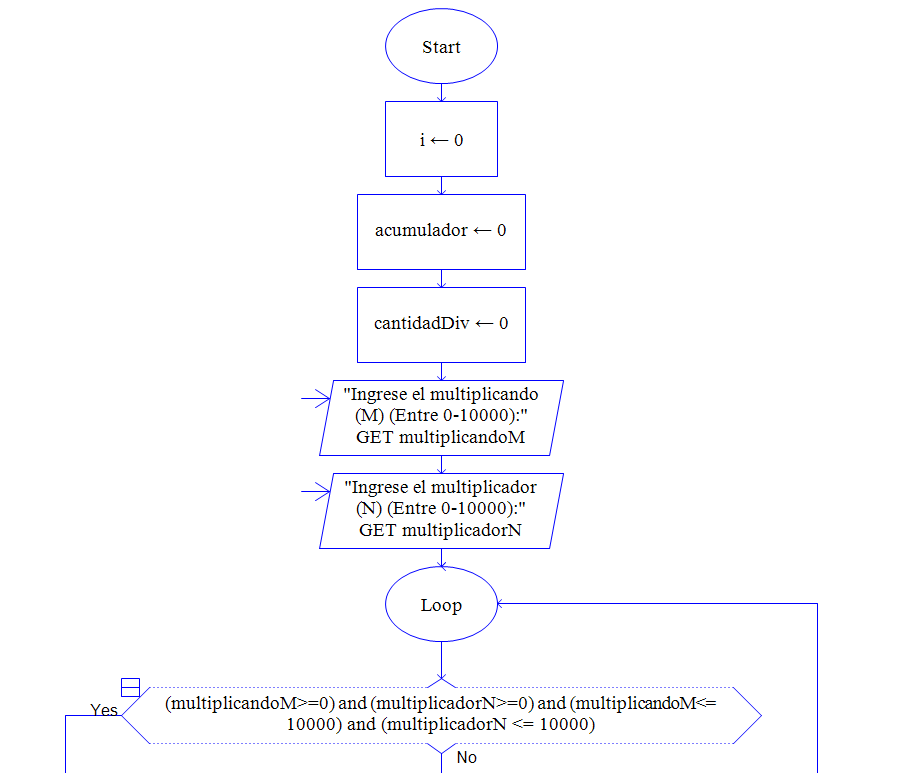
procesoN = procesoN\*2;

Fin Para

Mostrar "El resultado de multiplicar ", multiplicandoM " x ", multiplicadorN, " es: ", acumulador;

FinAlgoritmo

## 3) Diagrama de Flujo:



## 4) Código en C:

#include <stdio.h>

#include <math.h>

#include <locale.h>

#include <conio.h>

#include <time.h>

int main(int *argc*, char const \**argv*[])

{

    setlocale(LC\_CTYPE, "es\_ES.UTF-8");

    int multiplicandoM, multiplicadorN, procesoM, procesoN, acumulador, cantidadDiv, i;

*clock\_t* start, end;

    double cpu\_time\_used;

    multiplicandoM = 0;

    multiplicadorN = 0;

    procesoM = 0;

    procesoN = 0;

    acumulador = 0;

    cantidadDiv = 0;

    printf("Ingrese el multiplicando (M) (Entre 0-10000):\n");

    scanf("%i", &multiplicandoM);

    printf("Ingrese el multiplicador (N) (Entre 0-10000):\n");

    scanf("%i", &multiplicadorN);

    while (multiplicandoM<0 || multiplicadorN<0 || multiplicandoM>10000 || multiplicadorN>10000)

    {

        printf("Al menos un número ingresado está fuera del rango definido o tiene decimales\n");

        printf("Ingrese números enteros positivos dentro del rango\n");

        printf("Ingrese el multiplicando (M) (Entre 0-10000):\n");

        scanf("%i", &multiplicandoM);

        printf("Ingrese el multiplicador (N) (Entre 0-10000):\n");

        scanf("%i", &multiplicadorN);

    }

    //--------------------------------------START TIME--------------------------------------

    start = clock();

    //--------------------------------------END TIME--------------------------------------

    procesoM = multiplicandoM;

    procesoN = multiplicadorN;

    while (procesoM>0)

    {

        cantidadDiv = cantidadDiv+1;

        procesoM = trunc(procesoM/2);

    }

    procesoM = multiplicandoM;

    for (i = 0; i <= cantidadDiv; i++)

    {

        printf("M: %i\n", procesoM);

        if ((procesoM%2)==1)

        {

            acumulador = acumulador + procesoN;

            printf("N: %i\*\n", procesoN);

        }

        else

        {

            printf("N: %i\n", procesoN);

        }

        printf("Acumulado: %i\n", acumulador);

        procesoM = trunc(procesoM/2);

        procesoN = procesoN\*2;

    }

    printf("El resultado de multiplicar %i x %i, es: %i\n", multiplicandoM,multiplicadorN,acumulador);

    //--------------------------------------START TIME--------------------------------------

    end = clock();

    cpu\_time\_used = ((double) (end - start)) / CLOCKS\_PER\_SEC;

    printf("Tiempo de ejecución: %f\n", cpu\_time\_used);

    //--------------------------------------END TIME--------------------------------------

    getch();

    return 0;

}

## 5) Pruebas de Escritorio:





# CUADRO COMPARATIVO:

|  |  |  |
| --- | --- | --- |
|  | **Solución 1** | **Solución 2** |
| **Cantidad de variables** | 5 | 7 |
| **Tiempo de ejecución (Solo proceso)** |  |  |
| **Tiempo de ejecución (Todo el código)** |  |  |
| **Cantidad de líneas del código** | 84 | 87 |
| **Claridad en nombre de variables** | Variables no tan descriptivas | Variables descriptivas |

Se escoge la solución 1 frente a la solución 2, por que la 1 tiene más puntos positivos que benefician a la ejecución del código.

# Conclusiones:

En conclusión, se elige la solución 1 frente a la 2, debido a que en la solución 1 la cantidad de variables es menor. Asimismo, el tiempo de ejecución y la cantidad de líneas de código es menor frente a la segunda solución. A pesar de ello, se puede observar una mayor claridad en el nombre de variables del segundo código. Es por ello, que, si se observa en el código, las variables son más específicas y descriptivas.

# LINK GITHUB:

<https://github.com/emilia6cano/proyecto-integrador>