

Universidad de las Américas (UDLA)

Integrantes: Anahí Mancero, Leandro Pozo
Telecomunicaciones

Materia: Algoritmos

Carrera:

Fecha: 02/02/2024

Ejercicios Prácticos

Objetivos:

- 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.
- Los estudiantes deben analizar el problema planteado, definir claramente los datos y procesos involucrados en los problemas, proponer y evaluar la mejor alternativa para resolver los problemas, 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, además del pseudocódigo y las pruebas de escritorio respectivas, y realizar el ejercicio en C.

Problemas:

1) Al interior de una organización el director general ha establecido tener reuniones masivas con todos los empleados de la planta en primer lunes de cada trimestre. Elabore un algoritmo que dado el año calcule automáticamente las fechas de las reuniones trimestrales de todo el año. Use las siguientes fórmulas para determinar el día de la semana de una fecha día/mes/año, el valor 0 significa domingo, 1 lunes y así sucesivamente.

$$a = (14 - \text{mes}) / 12$$

$$y = \text{anio} - a$$

$$m = \text{mes} + 12 * a - 2$$

$$d = (\text{dia} + y + y/4 - y/100 + y/400 + 31 * m/12) \bmod 7$$

2) 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.

Solución problema 1:

ALTERNATIVAS DE SOLUCIÓN

Solución con Estructura for:

```
//ENTRADA
```

```
int main() {
```

```
    int year;
```

```
    std::cout << "Ingrese el año: ";
```

```

std::cin >> year;

//PROCESO
for (int i = 1; i <= 12; i += 3) {
    int day = (14 - i) / 12;
    int month = i + 12 * day - 2;
    int dayOfWeek = (year + year/4 - year/100 + year/400 + 31 * month/12) % 7;
    //SALIDA
    std::cout << "La reuniÃ³n trimestral en " << month << "/" << year << " cae en el dia " <<
    dayOfWeek << " del trimestre " << i/3 << "." << std::endl;
}
return 0;
}

```

Pseudocodigo (Pseint):

Algoritmo ReunionesTrimestrales

 //Entrada

 Definir anio, a, h, m, d Como Entero

 //Proceso

 Escribir "Ingrese el aÃ±o:"

 Leer anio

 Para mes Desde 1 Hasta 12 Con Paso 3

 a = (14 - mes) / 12

 h = anio - a

 m = mes + 12 * a - 2

 d = (1 + h + h / 4 - h / 100 + h / 400 + 31 * m / 12) Mod 7

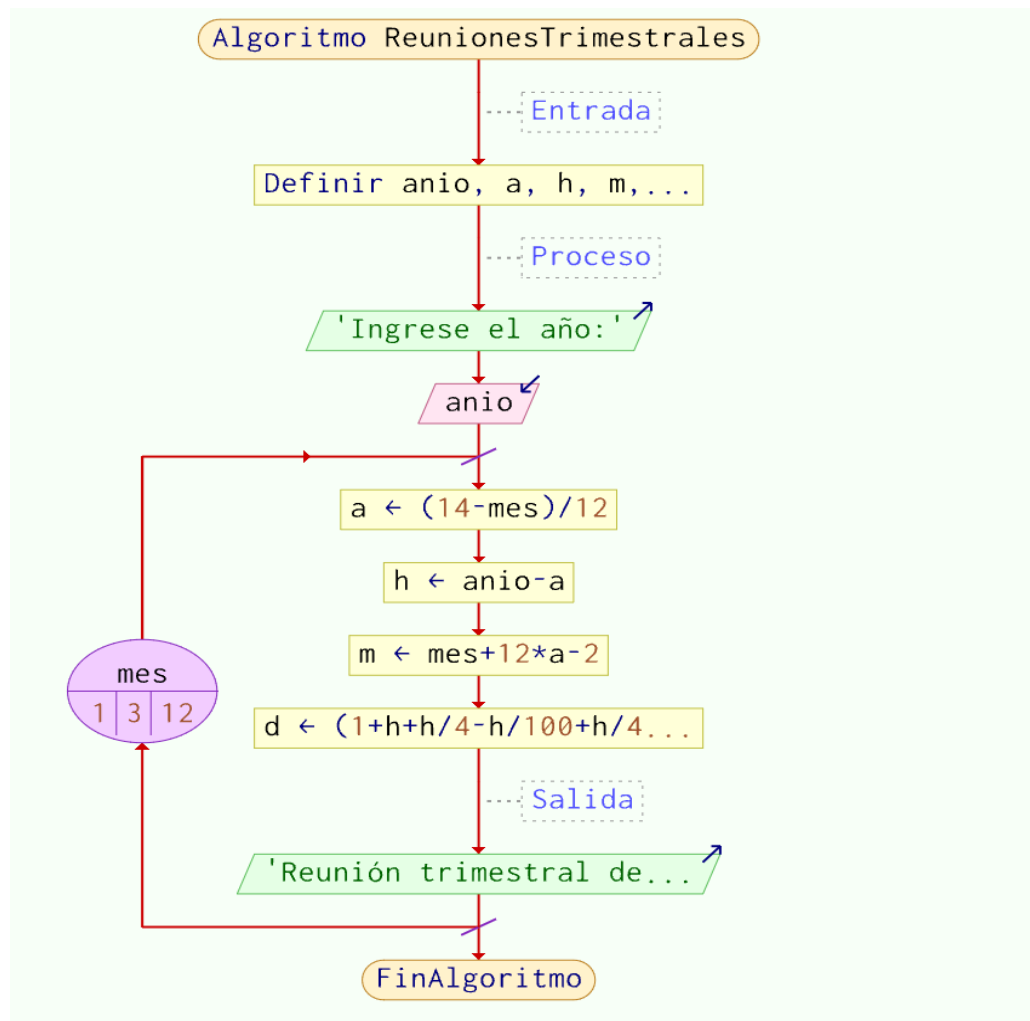
 //Salida

 Escribir "ReuniÃ³n trimestral del ", mes, "/", 1, "/", anio, " serÃ¡ en el dÃ­a ", d

 Fin Para

Fin Algoritmo

Diagrama de Flujo



Solución con la estructura “While”

```

#include <iostream>

//Entrada
int main() {
    int anio, a, y, m, d;

    std::cout << "Ingrese el año: ";
    std::cin >> anio;
    m = 1; // Inicializar mes en 1

    //Proceso
    while (m <= 12) {
        a = (14 - m) / 12;
        y = anio - a;
    }
}

```

```

        d = (1 + y + y / 4 - y / 100 + y / 400 + 31 * m / 12) % 7;
//Salida

        std::cout << "Reunión trimestral del " << m << "/" << anio << " será en el día " << d <<
std::endl;

        m += 3; // Incrementar mes en 3 para el siguiente trimestre
    }

    return 0;
}

```

Pseudocodigo (Pseint):

Algoritmo ReunionesTrimestrales

```

        //Entrada

        Definir anio, a, h, m, d Como Entero

        Escribir "Ingrese el año:"

        Leer anio

        m = 1 // Inicializar mes en 1

        //Proceso

        Mientras m <= 12 Hacer

            a = (14 - m) / 12

            h = anio - a

            d = (1 + h + h / 4 - h / 100 + h / 400 + 31 * m / 12) Mod 7

            //Salida

            Escribir "Reunión trimestral del ", m, "/", anio, " será en el día ", d

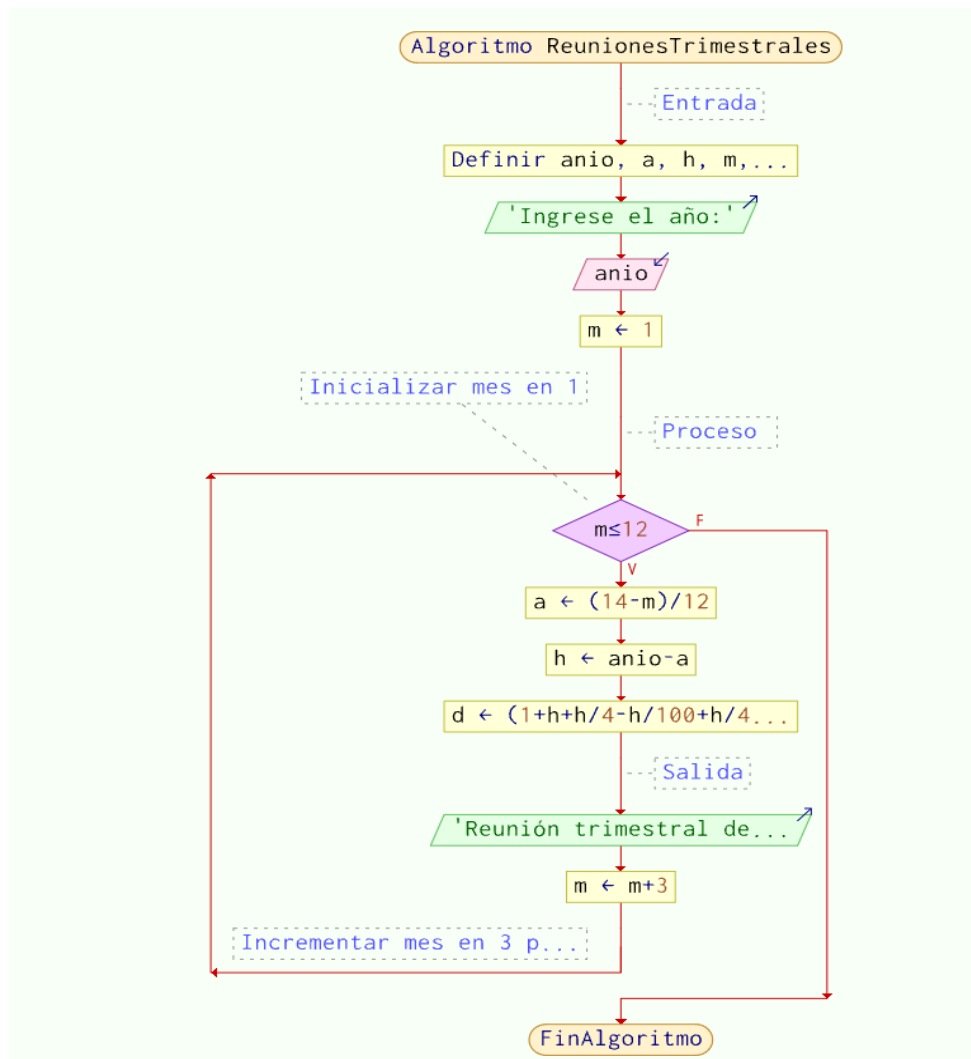
            m = m + 3 // Incrementar mes en 3 para el siguiente trimestre

        Fin Mientras

    Fin Algoritmo

```

Diagrama de Flujo



La Mejor Opción:

La opción más optima a usar es con la estructura "For " porque La estructura for es apropiada porque hay un número específico y conocido de iteraciones (doce en este caso, una por cada mes). La variable i se incrementa en cada iteración con una secuencia fija (i += 3), lo que indica claramente que el bucle se ejecutará exactamente 12 veces.

Un bucle for es generalmente más legible y conciso cuando se sabe de antemano cuántas iteraciones se realizarán. La inicialización, la condición y la actualización se colocan en la misma línea, lo que facilita la comprensión del comportamiento del bucle.

Presentación:

https://www.canva.com/design/DAF7rN83WOW/sbZ2xLh1ukJuEsP-5uNbXg/edit?utm_content=DAF7rN83WOW&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Pruebas de Escritorio:

```
C:\Users\jorda\AppData\Local x + v
Ingrese el año: 22
La reunión trimestral en 2 / 22 cae en el día 4 del trimestre 1.
La reunión trimestral en 5 / 22 cae en el día 4 del trimestre 2.
La reunión trimestral en 8 / 22 cae en el día 5 del trimestre 3.
La reunión trimestral en 11 / 22 cae en el día 6 del trimestre 4.

-----
Process exited with return value 0
Press any key to continue . . .
```

```
C:\Users\jorda\AppData\Local x + v
Ingrese el año: 23
La reunión trimestral en 2 / 23 cae en el día 5 del trimestre 1.
La reunión trimestral en 5 / 23 cae en el día 5 del trimestre 2.
La reunión trimestral en 8 / 23 cae en el día 6 del trimestre 3.
La reunión trimestral en 11 / 23 cae en el día 0 del trimestre 4.

-----
Process exited with return value 0
Press any key to continue . . .
```

Solución problema 2:

Solución con la Estructura for:

```
#include <iostream>

// ENTRADA: Función que realiza la multiplicación rusa con bucle for
int multiplicacionRusa(int numeroMenor, int numeroMayor) {

    int suma = 0;

    // PROCESO: Mostrar encabezado para mostrar resultados
    std::cout << "NUMERO MENOR \t NUMERO MAYOR \t NUMERO MENOR IMPAR \t SUMA" <<
std::endl;

    // PROCESO: Bucle para realizar la multiplicación rusa
    for (; numeroMenor > 0; numeroMenor /= 2, numeroMayor *= 2) {

        // PROCESO: Verificar si numeroMenor es impar y acumular valor
        bool esImpar = (numeroMenor % 2 != 0);

        if (esImpar) {

            suma += numeroMayor;

        }

        // PROCESO: Mostrar valores en cada paso
        std::cout << numeroMenor << "\t\t" << numeroMayor << "\t\t" << (esImpar ? "sÍ-" : "no")
<< "\t\t\t" << suma << std::endl;

    }

    // PROCESO: Retornar el resultado final
    return suma;

}

// PROCESO: Solicitar al usuario los valores de numeroMenor y numeroMayor
int main() {

    int numeroMenor, numeroMayor;

    // PROCESO: Solicitar al usuario los valores de numeroMenor y numeroMayor
    std::cout << "Ingrese el valor del NÚMERO MAYOR: ";
```

```

std::cin >> numeroMayor;

std::cout << "Ingrese el valor del Número MENOR: ";

std::cin >> numeroMenor;

// PROCESO: Calcular el producto utilizando la función multiplicacionRusa
int resultado = multiplicacionRusa(numeroMenor, numeroMayor);

// SALIDA: Mostrar el resultado final

std::cout << "\nEl resultado final de " << numeroMenor << " * " << numeroMayor << " es: "
<< resultado << std::endl;

return 0;
}

```

Pseudocodigo (Pseint):

//Entrada

Proceso multiplicacionRusa

Definir numeroMenor, numeroMayor, suma, esImpar Como Entero

Escribir "Ingrese el valor del Número MAYOR: "

Leer numeroMayor

Escribir "Ingrese el valor del Número MENOR: "

Leer numeroMenor

Escribir "NUMERO MENOR NUMERO MAYOR NUMERO MENOR IMPAR SUMA"

 //Proceso

 esImpa = (numeroMenor Mod 2 <> 0)

 Si esImpa Entonces

 suma = suma + numeroMayor

 FinSi

 si numeroMenor % 2 <> 0 Entonces

 Escribir " si"

 SiNo

 Escribir " No"

FinSi

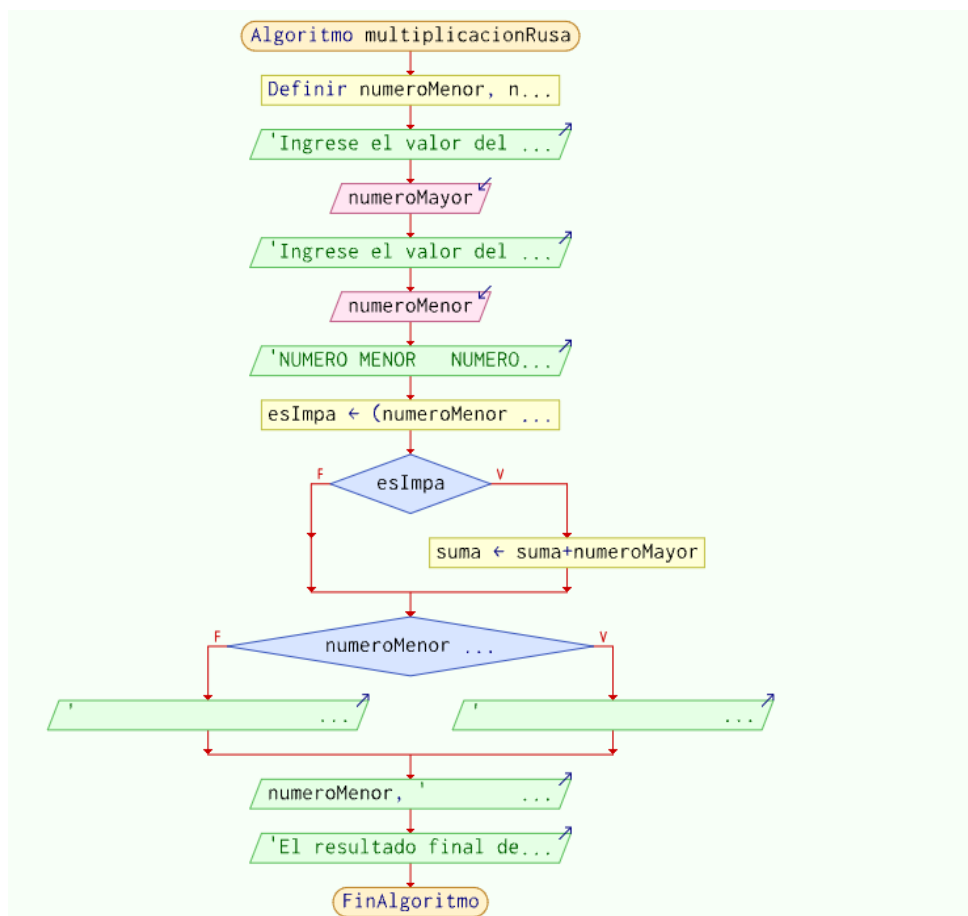
//Salida

Escribir numeroMenor, " ", numeroMayor, " " " ", suma

Escribir "El resultado final de ", numeroMenor, " * ", numeroMayor, " es: ",
numeroMenor* numeroMayor

FinProceso

Diagrama de Flujo:



Solución con la Estructura While:

//Incluye la biblioteca de entrada y salida de datos

#include <iostream>

```

int main() {

    // Entrada

    double numeroMayor, numeroMenor;

    std::cout << "Ingrese el valor del Número MAYOR: ";

    std::cin >> numeroMayor;

    std::cout << "Ingrese el valor del Número MENOR: ";

    std::cin >> numeroMenor;


    std::cout <<
    "=====\\n"
    ;

    std::cout << "                El resultado de la multiplicación es:" << numeroMenor << " * "
    << numeroMayor

        << " es: " << numeroMayor * numeroMenor << "\\n";

    std::cout << "
    =====\\n";

    std::cout << "                NUMERO MENOR IMPAR  NUMERO MENOR  NUMERO MAYOR
    SUMA\\n";

    std::cout << "                -----\\n";


    double suma = 0;


    // Proceso

    while (numeroMenor > 0) {

        bool esImpar = (static_cast<int>(numeroMenor) % 2 != 0);

        if (esImpar) {

            suma += numeroMayor;

        }


        std::cout << (esImpar ? "                sí" : "                no");

```

```
std::cout << "          " << numeroMenor << "          " << numeroMayor << "
" << suma << "\n";
```

```
// Salida

numeroMenor = static_cast<int>(numeroMenor / 2);

numeroMayor *= 2;

}

return 0;

}
```

Pseudocodigo (Pseint):

Algoritmo MultiplicacionRusa

```
//Entrada

Definir numeroMayor, numeroMenor Como Real

Escribir "Ingrese el valor del Número MAYOR: "

Leer numeroMayor

Escribir "Ingrese el valor del Número MENOR: "

Leer numeroMenor

    escribir"=====
=====

    Escribir "El resultado de la multiplicacion es ", numeroMenor, " * ", numeroMayor, "
es: ", numeroMayor*numeroMenor

    Escribir
"=====

Escribir "NUMERO MENOR  NUMERO MAYOR  NUMERO MENOR IMPAR    SUMA"

Escribir "-----"

    suma <- 0

//Proceso

    Mientras numeroMenor > 0 Hacer

        esImpar <- (numeroMenor % 2 <> 0)

        Si esImpar Entonces

            suma <- suma + numeroMayor

        FinSi
```

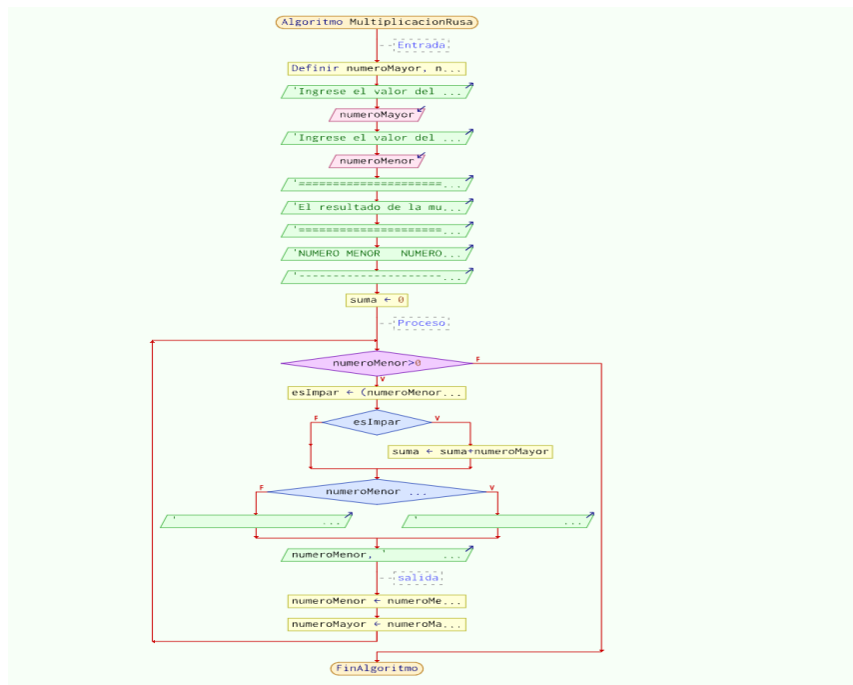
```

    escribir "          sí"
  SiNo
    escribir"          no"
  FinSi
Escribir numeroMenor, "          " numeroMayor "          " " "          " suma

```

FinAlgoritmo

Diagrama de Flujo:



LA Mejor Opción:

La opción más optima a usar es con la estructura “While” porque, el código realiza una multiplicación de dos números y luego realiza un bucle while para sumar el valor del número mayor cuando el número menor es impar, dividiendo el número menor por 2 en cada iteración hasta que sea 0. La elección del bucle while es apropiada en este caso porque la condición de continuación está basada en el valor de numeroMenor, no en un contador explícito.

PRUEBAS DE ESCRITORIO

```
C:\Users\jorda\AppData\Local... x + v
Ingrese el valor del N||mero MAYOR: 24
Ingrese el valor del N||mero MENOR: 12
NUMERO MENOR    NUMERO MAYOR    NUMERO MENOR IMPAR    SUMA
12              24              no                    0
6               48              no                    0
3               96              s-i                  96
1              192              s-i                  288

El resultado final de 12 * 24 es: 288

-----
Process exited with return value 0
Press any key to continue . . . |
```

```
C:\Users\jorda\AppData\Local... x + v - □ x
Ingrese el valor del N||mero MAYOR: 34
Ingrese el valor del N||mero MENOR: 12
NUMERO MENOR    NUMERO MAYOR    NUMERO MENOR IMPAR    SUMA
12              34              no                    0
6               68              no                    0
3              136              s-i                  136
1              272              s-i                  408

El resultado final de 12 * 34 es: 408

-----
Process exited with return value 0
Press any key to continue . . . |
```

