

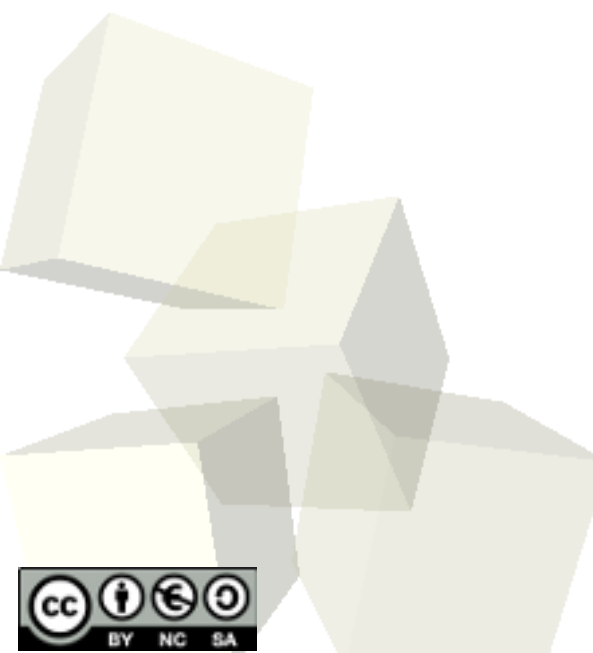


Tema 2

Algoritmos y programas Ejemplos para introducir las funciones

Informàtica
Grado en Física
Universitat de València

Francisco.Grimaldo@uv.es
Ariadna.Fuertes@uv.es





Programa: m.c.d. (1/4)

- Realizar un programa que calcule el máximo común divisor de dos números.
- **1º.- Análisis del problema:**
 - ♦ El m.c.d de dos o más números naturales es el mayor divisor posible de todos ellos.
 - ♦ Datos de entrada: 2 números (necesito 2 variables enteras para almacenarlos)
 - ♦ Datos de salida: mcd (otra variable entera)
- **2º.-Diseño/búsqueda del algoritmo**
 - ♦ Encontramos dos algoritmos:
 - Descomponer en factores primos y coger los comunes con su menor exponente.
 - Usar el algoritmo de Euclides que consiste en “divisiones sucesivas”...



■ El algoritmo de Euclides:

Dados dos números, hacer:

- ♦ En la primera división se toma como dividendo el mayor de los números y como divisor el otro.
- ♦ Después, el divisor y el resto sirven de dividendo y divisor en la siguiente división.
- ♦ El proceso se para cuando se obtiene un resto nulo.
- ♦ Y el mcd es el penúltimo resto.

Vamos a implementar el algoritmo en dos versiones; usando el “operador resto” y sin usarlo (mediante restas).



Programa: m.c.d. (3/4)

Pseudocódigo:

Versión 1: operador resto

Variables:

a, b, r : enteros

Inicio

Hacer₁

Leer(a, b)

Mientras (a<=0 OR b<=0)

Si a>b entonces

r ← a%b

Sino

r ← b%a

Fin_Si

Hacer₂

a ← b

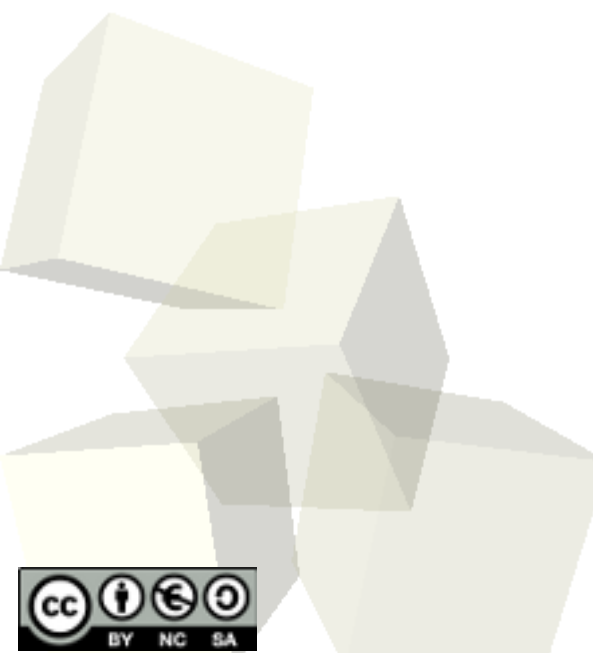
b ← r

r ← a%b

Mientras (r≠0)

Escribir (a)

FIN





Programa: m.c.d. (4/4)

Pseudocódigo:

Versión 2: con restas

Variables:

a, b : enteros

Inicio

Hacer₁

Leer(a, b)

Mientras ($a \leq 0$ OR $b \leq 0$)

Mientras ($a \neq b$) hacer

Si $a > b$ entonces

$a \leftarrow a - b$

Sino

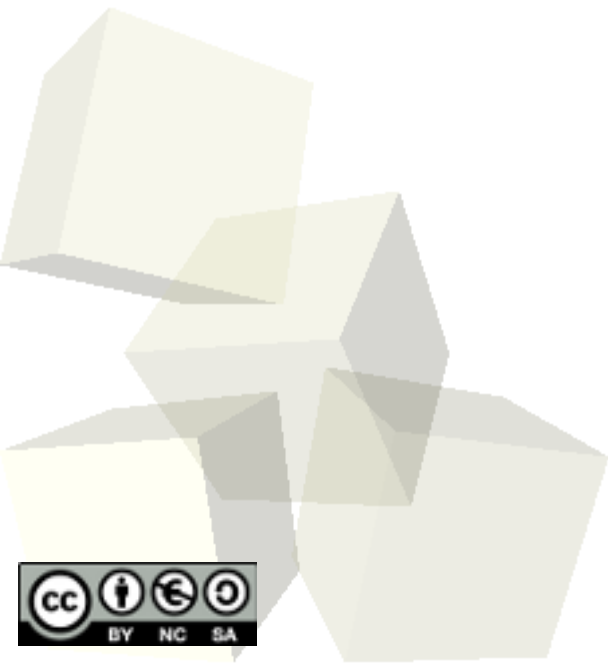
$b \leftarrow b - a$

Fin_Si

Fin_Mientras

Escribir (a)

FIN



Programa: m.c.d. de N números(1/3)

- Realizar un programa que calcule el máximo común divisor de N números introducidos por teclado.
- **1º.- Análisis del problema:**
 - ♦ Aplicar el m.c.d de dos en dos de todos los números que introduzcan.
 - ♦ Datos de entrada: 2 variables enteras para dos números a comparar y 1 variable entera que me dice cuántos números voy a comparar
 - ♦ Datos de salida: 1 variable entera que da el mcd de todos
- **2º.-Diseño/búsqueda del algoritmo**
 - ♦ Usaremos la versión 2 del algoritmo de Euclides aplicándola de dos en dos.

Programa: m.c.d. de N números(2/3)

Pseudocódigo:

Variables:

a, b, N, i : enteros

Inicio

Hacer₁

Leer(N,a, b)

Mientras₁ (a<=0 OR b<=0 OR N<2)

i ← 2

Hacer₂

Mientras (a ≠ b) hacer

Si a>b entonces

a ← a - b

Sino

b ← b - a

Fin_Si

Fin_Mientras

Hacer₃

Leer (b)

Mientras₃ (b<=0)

i ← i + 1

Mientras₂ (i <= N)

Escribir (a)

FIN

Módulo que resuelve el algoritmo de Euclides para 2 números → Vamos a hacer una Función para ello.
Y un procedimiento para la lectura de números

Programa: m.c.d. de N números(3/3)

Pseudocódigo del programa modularizado:

FUNCION Fun_mcd (a:entero, b:entero) : entero

Inicio

Mientras ($a \neq b$) hacer

Si $a > b$ entonces

$a \leftarrow a - b$

Sino

$b \leftarrow b - a$

Fin_Si

Fin_Mientras

Fun_mcd $\leftarrow a$

FIN_FUNCION

PROCEDIMIENTO Proc_Leer (ref x:entero)

Inicio

Hacer

Leer (x)

Mientras ($x \leq 0$)

FIN_PROCEDIMIENTO

Programa principal

Variables:

a, b, N, i : enteros

Inicio

Leer(N)

Si $N > 2$ entonces

Proc_Leer(a)

Proc_Leer(b)

$i \leftarrow 2$

Hacer

$a \leftarrow \text{Fun_mcd}(a, b)$

Proc_Leer (b)

$i \leftarrow i + 1$

Mientras ($i \leq N$)

Escribir (a)

Sino

Escribir("Num. Insuficientes")

FIN



Programa: M.C.M. (1/3)

- Realizar un programa que calcule el mínimo común múltiplo de dos números introducidos por teclado.
- **1º.- Análisis del problema:**
 - ♦ El M.C.M de dos o más números naturales es el menor número natural que es múltiplo de todos ellos.
 - ♦ Datos de entrada: 2 variables enteras
 - ♦ Datos de salida: 1 variable entera que da el MCM
- **2º.-Diseño/búsqueda del algoritmo**
 - ♦ Descomponer los números en factores primos y el MCM serán todos los factores comunes y no comunes con su mayor exponente.
 - ♦ Conociendo el m.c.d de dos números podemos decir

que:

$$\text{MCM}(a,b) = \frac{a * b}{\text{m.c.d.}(a,b)}$$



Programa: M.C.M. (2/3)

Pseudocódigo del programa modularizado:

FUNCION **Fun_mcd** (a:entero, b:entero) : entero

Inicio

Mientras ($a \neq b$) hacer

Si $a > b$ entonces

$a \leftarrow a - b$

Sino

$b \leftarrow b - a$

Fin_Si

Fin_Mientras

Fun_mcd $\leftarrow a$

FIN_FUNCION

PROCEDIMIENTO Proc_Leer (ref x:entero)

Inicio

Hacer

Leer (x)

Mientras ($x \leq 0$)

FIN_PROCEDIMIENTO

Programa principal del MCM

Variables:

a, b, res : enteros

Inicio

Proc_Leer(a)

Proc_Leer(b)

$res \leftarrow a * b / \text{Fun_mcd}(a, b)$

Escribir (res)

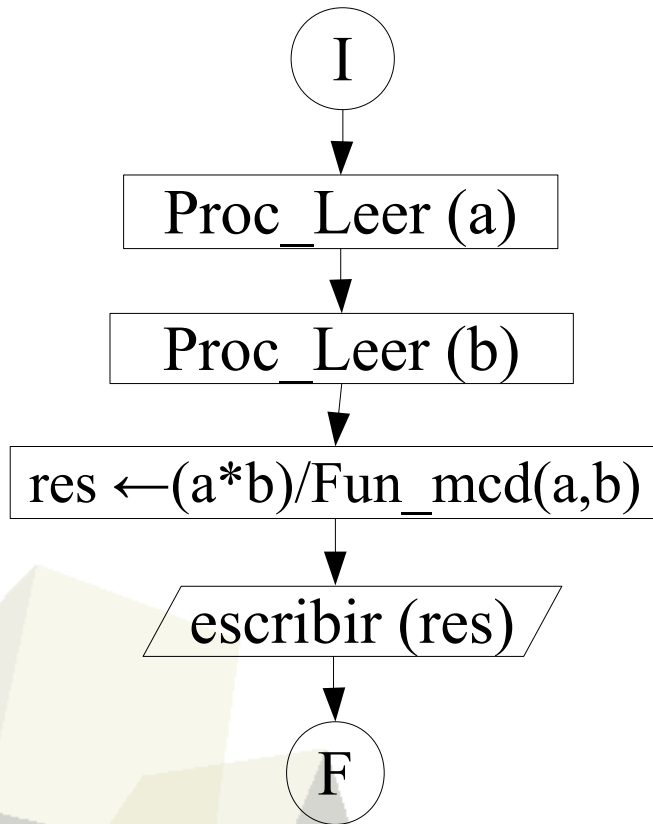
FIN



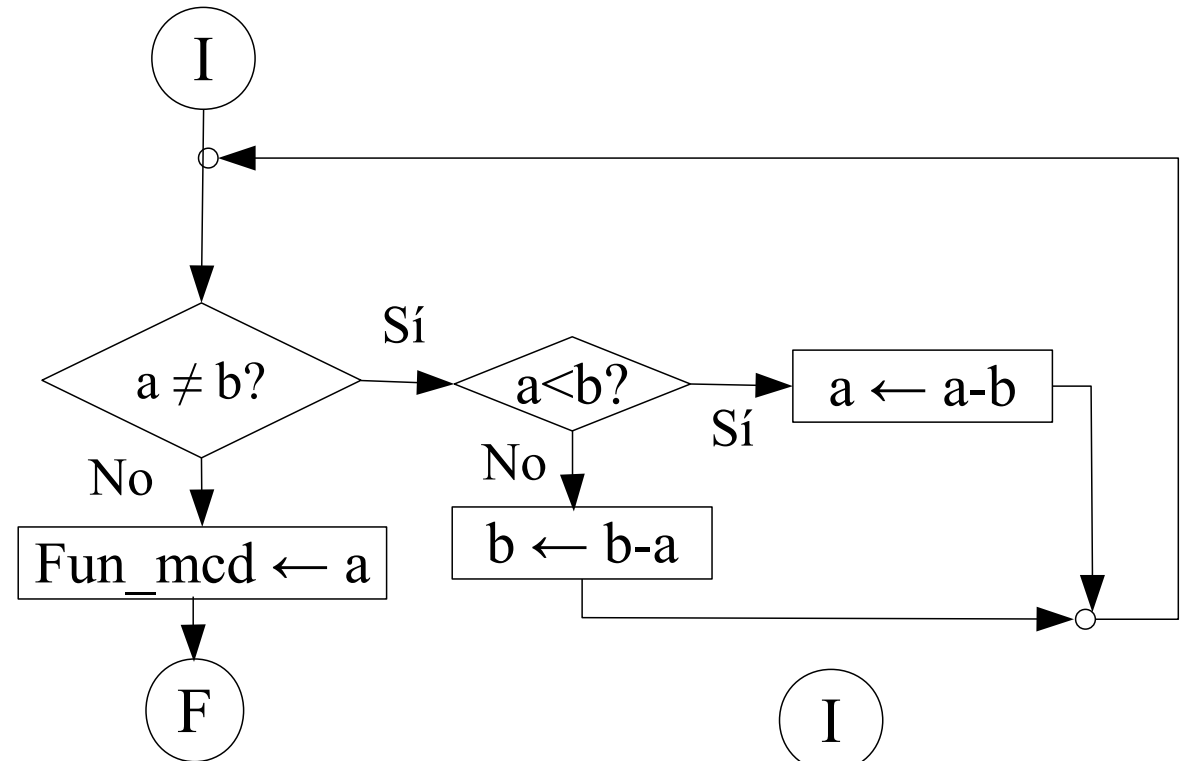
Programa: M.C.M. (3/3)

Organigramas:

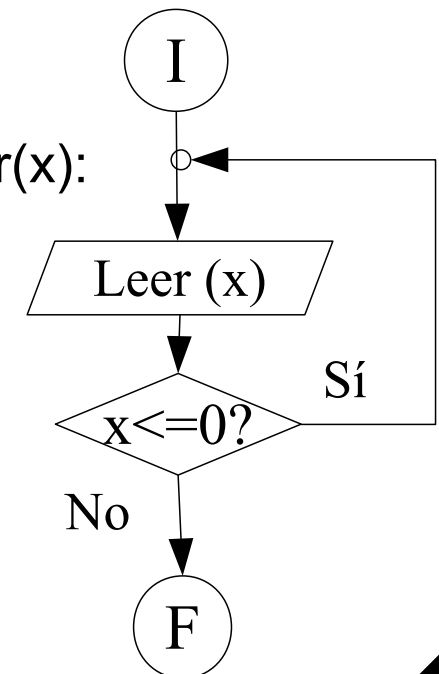
Programa principal:



Función Fun_mcd(a,b):



Procedimiento Proc_Leer(x):



Programa: Número Combinatorio (1/3)

- Escribid un programa que devuelva el número combinatorio de m sobre n . C_m^n
 - ♦ Datos de entrada: 2 números enteros (no negativos)
 - ♦ Datos de salida: 1 número entero
 - ♦ Búsqueda del algoritmo: “número de combinaciones posibles de m elementos tomados de n en n ”

$$C_m^n = C_{m,n} = C(m, n) = \binom{m}{n} = \frac{m!}{n!(m-n)!}$$

$$m > 0, n \geq 0 \text{ y } m \geq n$$

Programa: Número Combinatorio (2/3)

Pseudocódigo del programa modularizado:

FUNCION Fac (x : entero) : entero

Variables:

f, i : entero

Inicio

f ← 1

Desde i ← 2 hasta x hacer

f ← f * i

i ← i + 1

Fin_Desde

Fac ← f

FIN_FUNCION

PROCEDIMIENTO Proc_Leer (ref x:entero)

Inicio

Hacer

Leer (x)

Mientras (x ≤ 0)

FIN_PROCEDIMIENTO

Programa principal del Núm. Combinatorio

Variables:

m, n, res : enteros

Inicio

Proc_Leer(m)

Proc_Leer(n)

Si₁ m ≥ n AND m > 0 entonces

res ← Fac(m) / (Fac(n)*Fac(m-n))

Sino

res ← -1

Fin_Si₁

Si₂ (res = -1)

Escribir (“No es posible”)

Sino

Escribir (res)

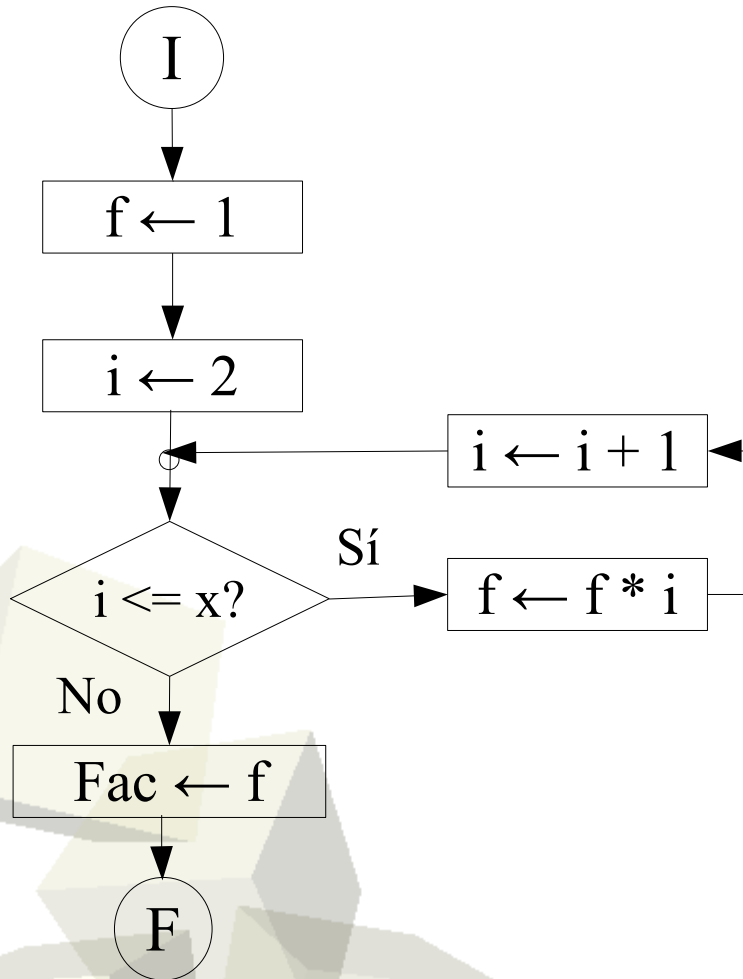
Fin_Si₂

FIN

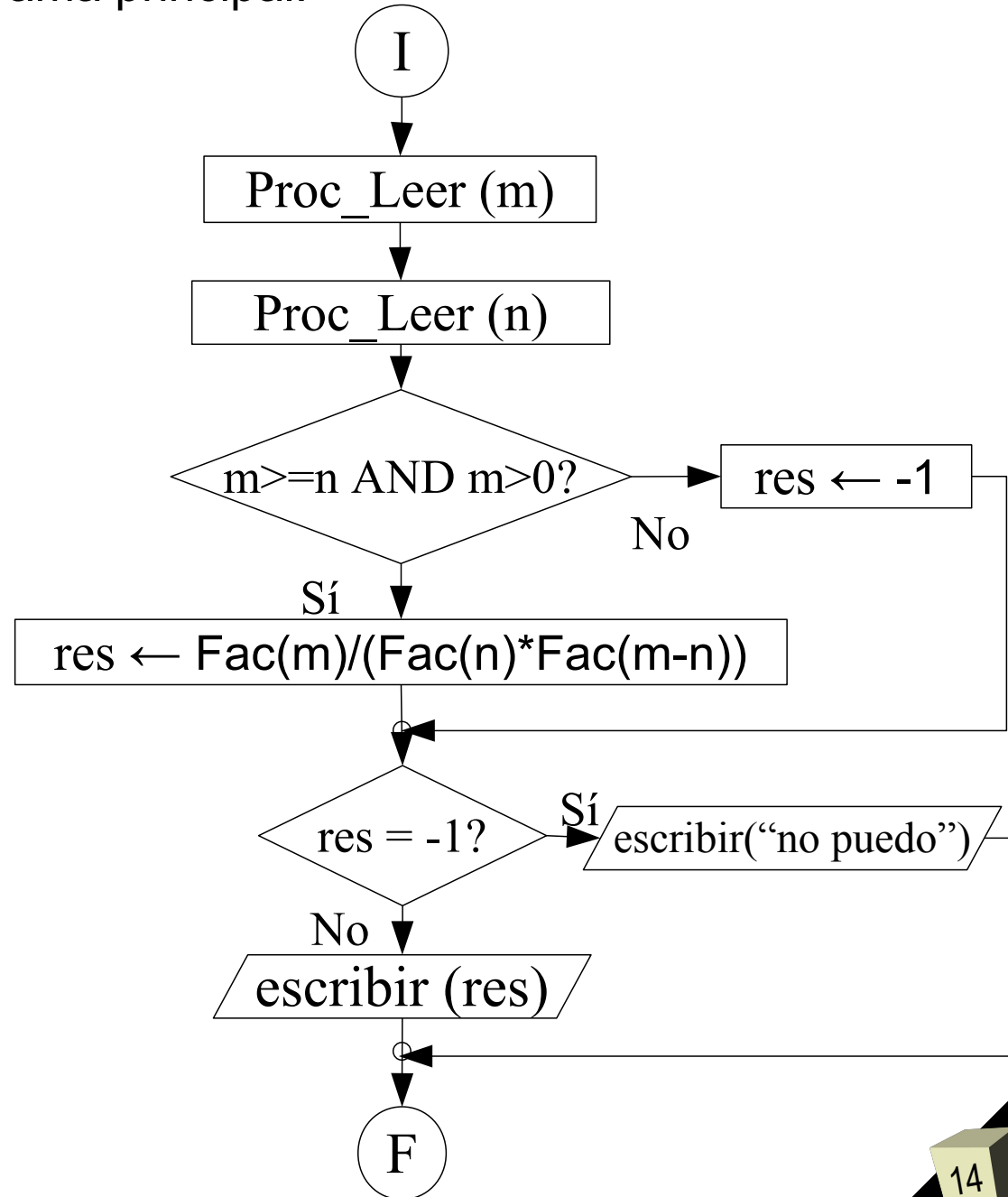
Programa: Número Combinatorio (3/3)

Organigramas:

Función Fac(x):



Programa principal:





- El valor de e^x se puede aproximar por la suma:

$$e^x = \sum_{i=0}^n \frac{x^i}{i!}$$

- Escribe un programa que le pida al usuario el valor de x y el valor de n y muestre por pantalla el valor de la aproximación de e^x para el x y el n introducidos.

- **1º.- Análisis del problema:**

- ♦ Datos de entrada: 2 variables enteras, x y n .
- ♦ Datos de salida: 1 variable real que da el valor de e^x
- ♦ Recordar: $e^0 = 1$ y para $x > 0$ podemos usar la fórmula.

- **2º.- Diseño/búsqueda del algoritmo**

- ♦ Si tenemos una función para la potencia y otra para el factorial, el programa se reduce a hacer el sumatorio.

$e=2.71828$



Pseudocódigo del programa modularizado:

FUNCION Fac (x : entero) : entero

Variables:

f, i : entero

Inicio

f ← 1

Desde i←2 hasta x hacer

f ← f * i

i ← i + 1

Fin_Desde

Fac ← f

FIN_FUNCION

FUNCION Pot (b: entero, e: entero) : entero

Variables:

res, i : entero

Inicio

res ← 1

Desde i←1 hasta e hacer

res ← res * b

i ← i + 1

Fin_Desde

Pot ← res

FIN_FUNCION

Programa principal de e^x

Variables:

x, n, i : enteros

sum :real

Inicio

Proc_Leer(x)

Proc_Leer(n)

sum ← 1

Desde i←1 hasta n hacer

sum ← sum + Pot(x,i) / Fac(i)

i ← i + 1

Fin_Desde

Si n=0 entonces

Escribir (“no hay aproximación”)

sino

Escribir (sum)

FIN

PROCEDIMIENTO Proc_Leer (ref x:entero)

Inicio

Hacer

Leer (x)

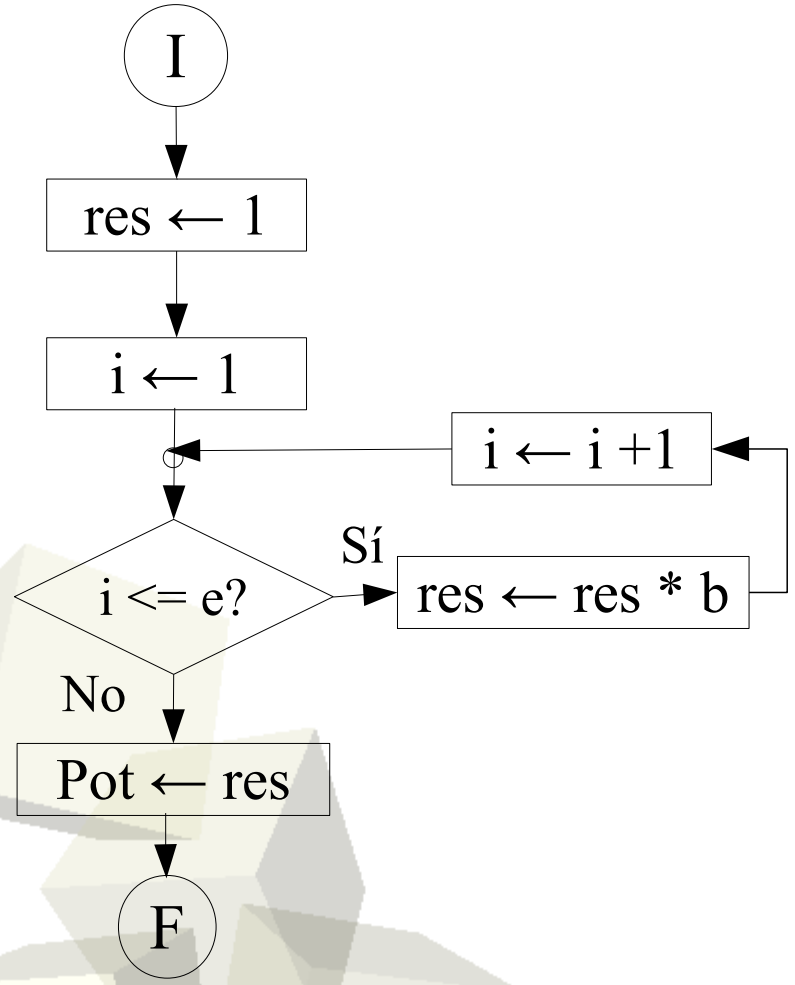
Mientras (x<0)

FIN_PROCEDIMIENTO

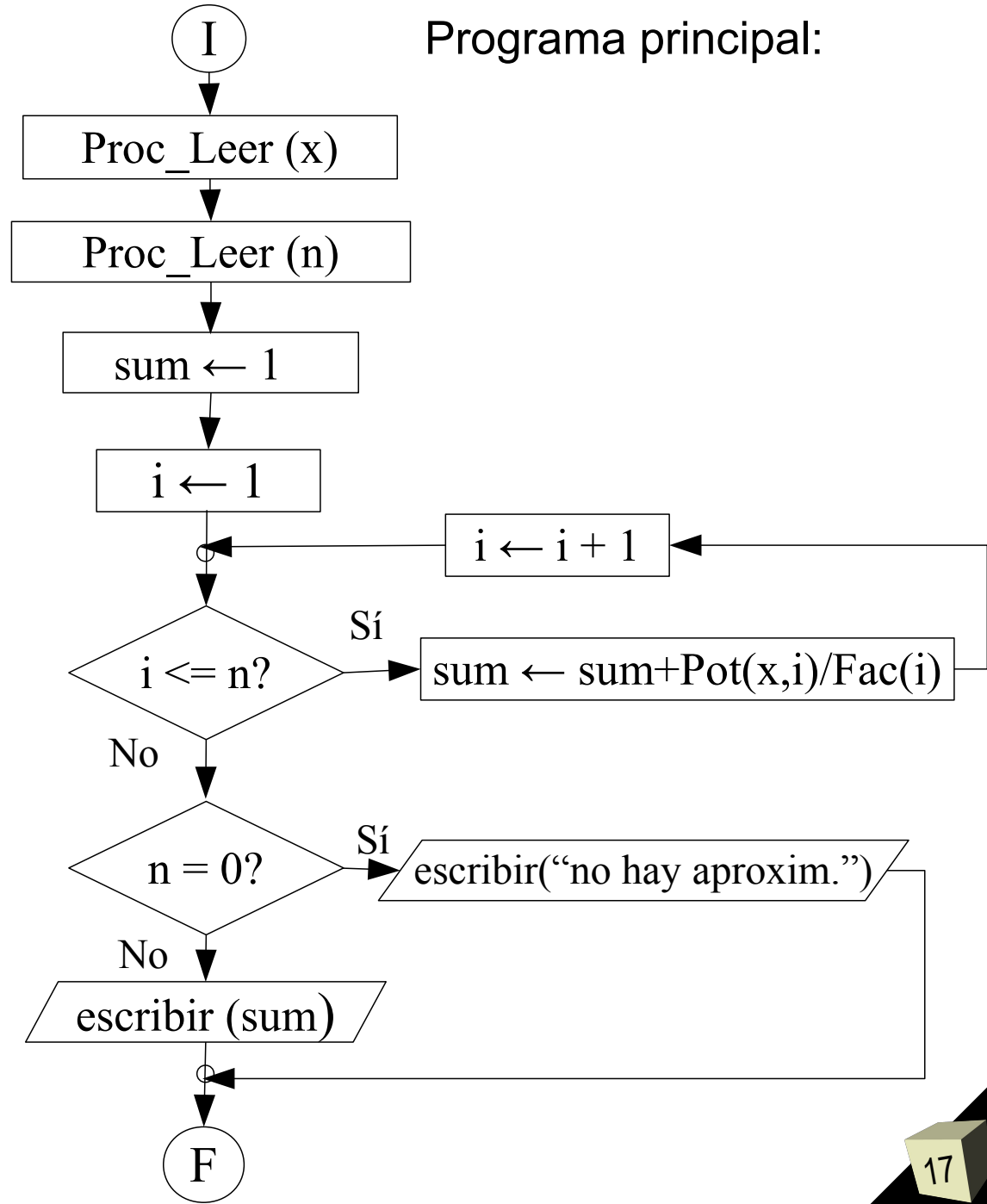


Organigramas:

Función Pot(b, e):



Programa principal:





Programa: Factorial como Procedimiento

Pseudocódigo del PROCEDIMIENTO Factorial:

PROCEDIMIENTO Fac (x : entero, ref. f : entero)

Variables:

i : entero

Inicio

f ← 1

Desde i ← 2 hasta x hacer

f ← f * i

i ← i + 1

Fin_Desde

FIN_PROCEDIMIENTO

Programa principal del factorial de un numero

Variables:

num, res : enteros

Inicio

Hacer

Leer (num)

Mientras (num < 0)

Fac(num, res)

Escribir (res)

FIN



Programa: Otras formas de programar el Factorial

Pseudocódigo de la función Factorial (bucle decreciente):

FUNCION **Fac** (x : entero) : entero

Variables:

f, i : entero

Inicio

f ← x

Desde i ← x-1 hasta 2 hacer

f ← f * i

i ← i - 1

Fin_Desde

Fac ← f

FIN_FUNCION

Pseudocódigo de la función Factorial en modo recursivo:

FUNCION **Fac** (x : entero) : entero

Variables:

f : entero

Inicio

f ← 1

Si x > 0 entonces

f ← **Fac**(x-1) * x

Fin_Si

Fac ← f

FIN_FUNCION