### Determine recursivamente si un número es Par o Impar usando sólo las operaciones suma y resta).

* *Como procedimiento:*

Procedimiento Espar(Num:entero) es

SI Num=0 entonces

Escribir (“El número es par”)

SINO

SI Num=1 entonces

Escribir (“El número es impar”)

SINO

EsPar(Num-2)

Fin\_Si

Fin\_Si

Fin\_Procedimiento

* *Como función:*

Función Espar(Num:entero: lógico es

SI Num=0 entonces

EsPar:=verdadero

SINO

SI Num=1 entonces

EsPar:=falso

SINO

EsPar:= EsPar(Num-2)//*MUY importante en funciones asignar a la misma la llamada recursiva*

Fin\_Si

Fin\_Si

Fin\_función.

function par(n : Integer) : Boolean;

begin

if n = 0 then par := true

else par := impar(n-1)

end;

function impar(n : Integer) : Boolean;

begin

if n = 0 then impar := false

else impar := par(n-1)

end;

implementación en C → <https://www.programiz.com/online-compiler/2rlvcieSn4yK8>

pueden correrlo directamente desde el compilador online y agregar “prints” si quieren ver como funciona cada paso detalladamente

### Dado un vector de 10 números enteros, calcular la suma de sus elementos.

Función Suma (vector: arreglo de (1..10) de enteros, n: entero): entero es

SI n=1 entonces

suma:=vector[n]

SINO

suma:= vector[n]+ suma(vector,n-1)

fin si

finfuncion

IMPLEMENTACION EN C → <https://www.programiz.com/online-compiler/8UmyBmWetZq3u>

### Leer una palabra (una cadena de caracteres) y la cantidad de caracteres y generar su palíndromo. El palíndromo de “Venezuela” es “aleuzeneV”.

Procedimiento Pal (palabra: arreglo 1…10 de caracter, n:entero) es

Algoritmo

Escribir (palabra(n));

Si n > 1 entonces

Pal(palabra, n-1);

Fsi;

Fprocedimiento;

*Aclaración necesaria para la definición de la función:* ***se debe llamar a la función con n = 10***

IMPLEMENTACION EN C → [***https://www.programiz.com/online-compiler/4gfqiyTztuGJV***](https://www.programiz.com/online-compiler/4gfqiyTztuGJV)

### Dada una lista de nombres ordenada en forma ascendente, construir un procedimiento recursivo que imprima como salida la misma lista, pero en orden descendente, sin modificar la lista original.

Funcion INVERTIR (P: puntero a Nodo) es

Si \*P.prox<>nil entonces entonces

INVERTIR (\*P.prox)

Fsi;

Escribir(\*P.dato)

FinFuncion.

*En esta función el caso base está implícito, será cuando no entre al condicional y solo escriba.*

### Construir un algoritmo recursivo que permita determinar si los dígitos de un número n dado son todos pares.

funcion todoDigitosPares(n:entero): booleano es

si n mod 2 <> 0 entonces

todoDigitosPares:= falso

sino

si n < 10 entonces

todoDigitosPares:= verdadero

sino

todoDigitosPares:= todoDigitosPares(n div 10)

fs

fs

ff

### Dados dos números enteros, divídalos (división entera) y muestre el resultado, usando sólo la operación resta.

Función divisionE(n1, n2: entero): entero es

Si (n1 < n2) entonces

divisionE:= 0;

sino

divisionE:= 1 + divisionE (n1-n2, n2)

Fsi;

Fin función

IMPLEMENTACION EN C →<https://www.programiz.com/online-compiler/8JeU6g5YC7uAa>

### Imprimir las cifras de un número n (siendo n≥0) en orden inverso al original. Por ej.: el inverso de 254 es 452.

procedimiento inverso(n:entero) es

si n < 10 entonces

esc(n)

sino

esc(n mod 10)

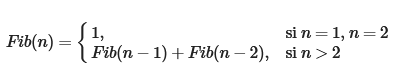
inverso(n div 10)

fs

fp

IMPLEMENTACION EN C → <https://www.programiz.com/online-compiler/2K0gkSvYo1NkC>

### Dado un número n como parámetro de entrada, calcular el n-ésimo número de la serie de Fibonacci. Tener en cuenta la siguiente definición:



Función Fibonacci(n: entero): entero es

si (n <= 1) entonces

Fibonacci := n

sino

Fibonacci := Fibonacci(n-1) + Fibonacci(n-2)

Fin si;

Fin función

### Dados dos números: a y b. Calcule la potencia a\*\*b, usando sólo multiplicaciones sucesivas.

Función Potencia (base, exponente: entero) es: entero  
Si exponente = 0 entonces  
 Potencia:= 1  
Sino  
 Potencia:= base \* Potencia(base, exponente -1)  
 Fsi;  
Fin función