# Recursión

- Concepto
  - Definición
    - Casos recursivos
      - Definiciones por inducción
      - Estructuras recursivas
- Ejemplo – Factorial
- Ejemplo de ejecución – Factorial(3)
- Ejemplo de ejecución – Imprimir datos de Lista
- Cuando Iteración y cuando Recursión
- Reglas para Iteración y Recursión

# Concepto de Recursión

Definición:

Repetición por autoreferencia, cuando un procedimiento o función se invoca a si mismo. (Puede ser en forma directa o indirecta)

Es la forma en la cual se especifica un proceso basado en su propia definición pero con menor complejidad, de esta forma en algún momento se llega a un proceso muy simple que puede resolverse fácilmente.

# Concepto de Recursión

**Ejemplos**:

1 - Definiciones por inducción:

*El cero pertenece a los Naturales, y si N pertenece a los Naturales N+1 también.*

2 – Estructuras recursivas:

*El nulo (nil) es una Lista y además un Nodo seguido de una Lista también es una Lista.*

# Ejemplo - Factorial

Definición por inducción:

*Factorial(0) = 1*
*Factorial(N) = N * Factorial (N-1)  para todo  N > 0*

# Ejemplo - Factorial

Definición :

*Factorial(0) = 1*

*Factorial(N) = N * Factorial (N-1)  para todo  N > 0*

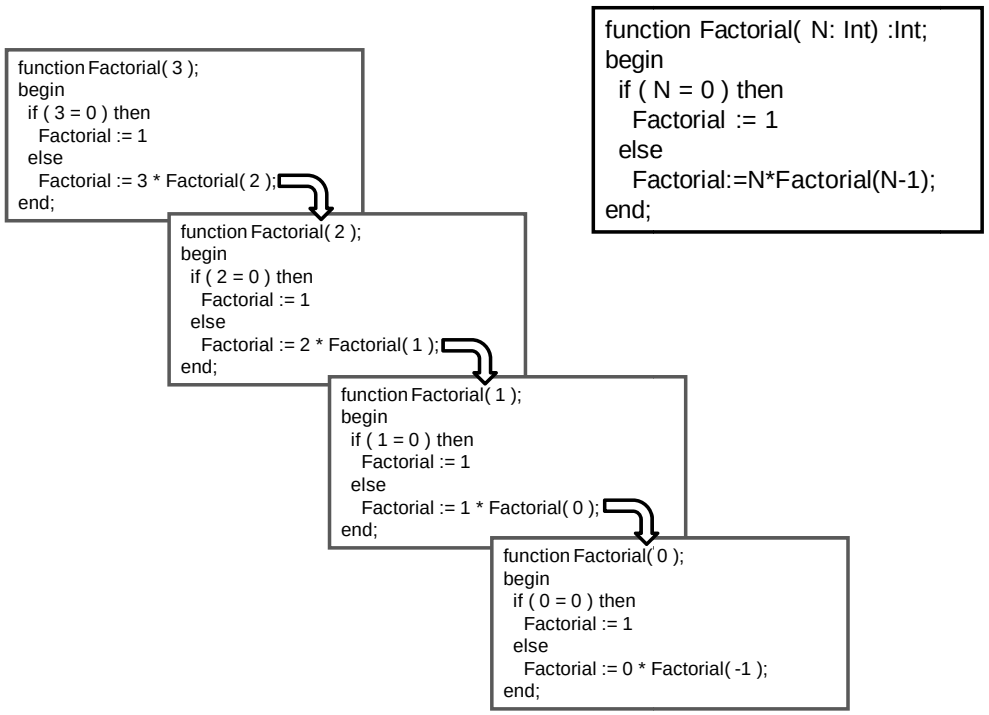
***Solución Iterativa:***

```
function FactIter( N: Int) :Int;
Var Fac : Integer;
Begin
  Fac := 1;
  if ( N > 0 ) then begin
    Fac := N;
    while (N > 1) do begin
      N := N – 1
      Fac := Fac * N;
    end;
  end;
  FactIter:=Fac;
end;
```

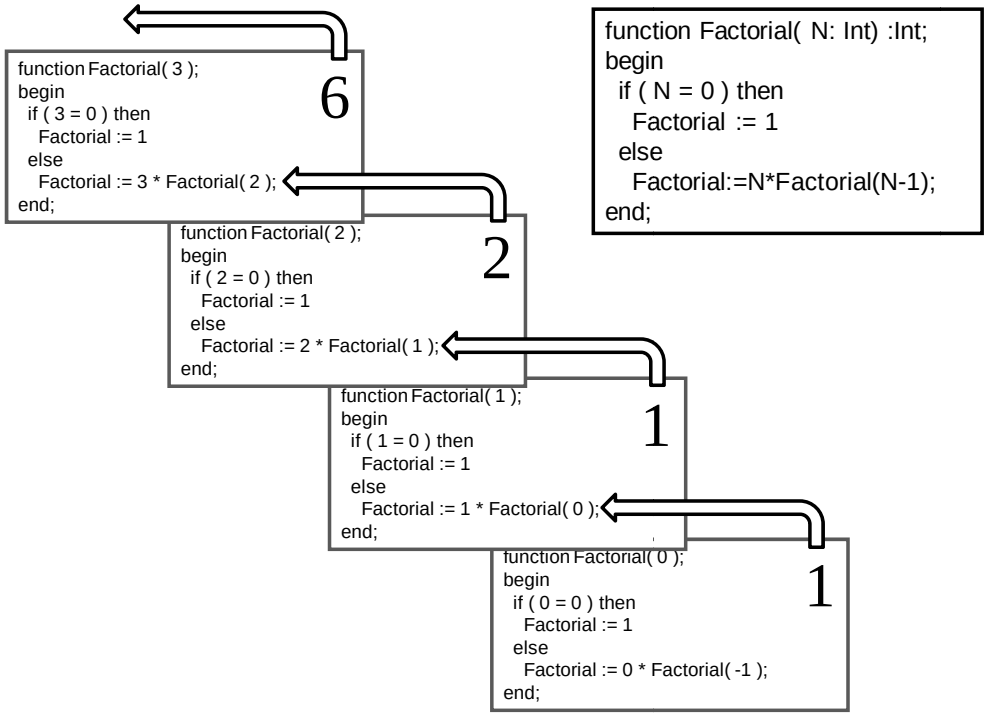
***Solución Recursiva:***

```
function Factorial( N: Int) :Int;
begin
  if ( N = 0 ) then
    Factorial := 1
  else
    Factorial:=N*Factorial(N-1);
end;
```

# Ejemplo: Factorial(3)




# Ejemplo: Factorial(3)

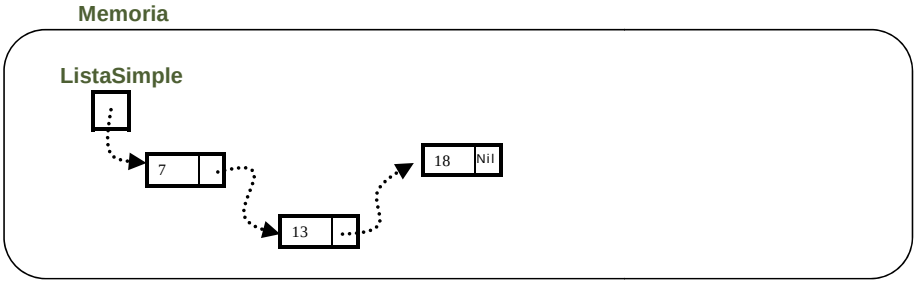


# Ejemplo: Factorial(3)

```
function Factorial( N: Int) :Int;
begin
  if ( N = 0 ) then
    Factorial := 1
  else
    Factorial:=N*Factorial(N-1);
end;
```

## Factorial(3) → 6

# Ejemplo: Recorrer lista con recursión



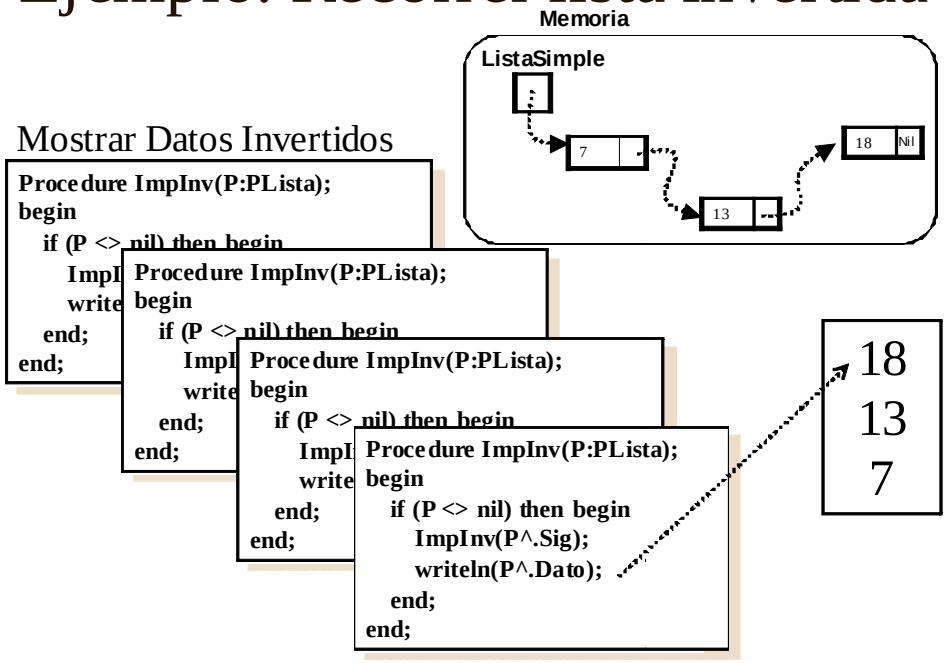

## Mostrar Datos Ordenados

```
Procedure ImpLista(P:PuntLista);
begin
   if (P <> nil) then begin
      writeln(P^.Dato);
      ImpLista(P^.Sig);
   end;
end;
```

## Mostrar Datos Invertidos

```
Procedure ImpListaInv(P:PuntLista);
begin
   if (P <> nil) then begin
      ImpListaInv(P^.Sig);
      writeln(P^.Dato);
   end;
end;
```

# Ejemplo: Recorrer lista invertida



# Cuándo usar Recursión

**Usar Recursión cuando**:

- ✔ *La estructura de la función es recurrente y el algoritmo resulta más sencillo.*
- ✔ *La estructura de datos es recursiva.*

**NO Usar Recursión cuando**:

- ✘ *Hay dificultad en la comprensión del algoritmo.*
- ✘ *Produce demandas excesivas de memoria o tiempo de ejecución.*

# Características a tener en cuenta

1. Hay una condición de corte con asignación de resultado. (Por este camino NO se invoca a la misma función)
2. Si no se da el corte, la próxima vez estaré más cerca. (Por este camino SI se invoca a la función)

***Iteración:***

```
function FactIter( N: Int) :Int;
Var Fac : Integer;
Begin
  Fac := 1;
  if ( N > 0 ) then begin
    Fac := N;
    while (N > 1) do begin
      N := N – 1
      Fac := Fac * N;
    end;
  end;
  FactIter:=Fac;
end;
```

***Recursión:***

```
function Factorial( N: Int) :Int;
begin
  if ( N = 0 ) then
    Factorial := 1
  else
    Factorial:=N*Factorial(N-1);
end;
```