



# Taller de Programación



# AGENDA



## Recursión



# AGENDA



Existen un conjunto de problemas que pueden resolverse siempre de la misma manera con la característica que el problema debe ir “achicandose” en cada instancia a resolver, hasta que en alguna instancia la solución es “trivial”.



La **recursividad** es una técnica de resolución de problemas que consiste en dividir un problema en instancias más pequeñas del mismo problema (también llamados subproblemas) hasta que obtengamos un subproblema lo suficientemente pequeño que tenga una solución trivial o directa.



# Recursión - MOTIVACION



## Contar las páginas de los siguientes libros

```
function cantidad (L:libros):integer;  
var  
    cantidadTotal:integer;  
  
Begin  
    cantidadTotal:=0;  
  
    mientras (hayaLibros)  
        cuentoPaginas de L  
        cantidadTotal:= cantidadTotal + cuentoPaginas  
  
    cantidad:= cantidadTotal;  
End;
```

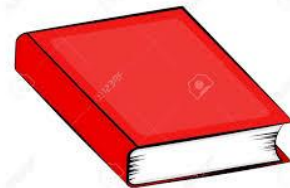
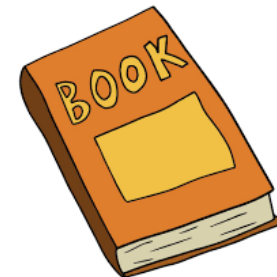
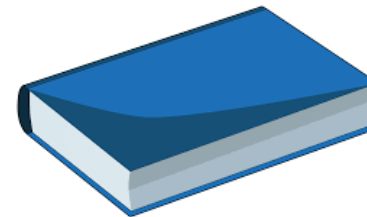
**Cantidad Total = 100**

30

15

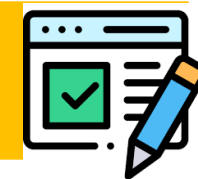
55

10





# Recursión - MOTIVACION

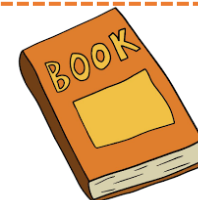
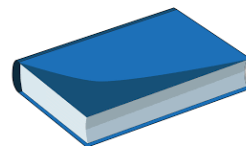


Contar las páginas de los siguientes libros

Cantidad Total = ~~1000~~

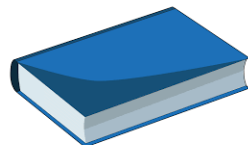


30

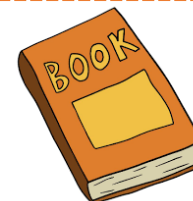


Cantidad Total

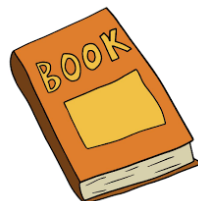
15



Cantidad Total

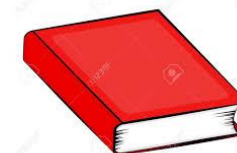
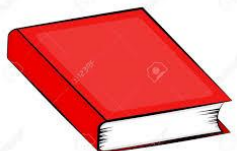


55



Cantidad Total

10

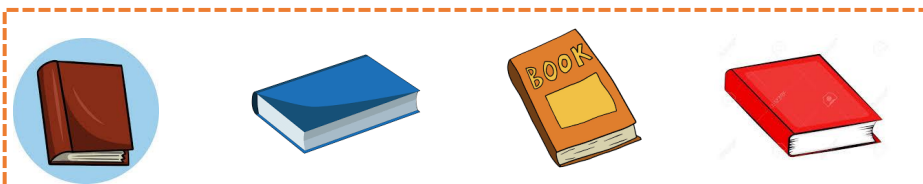
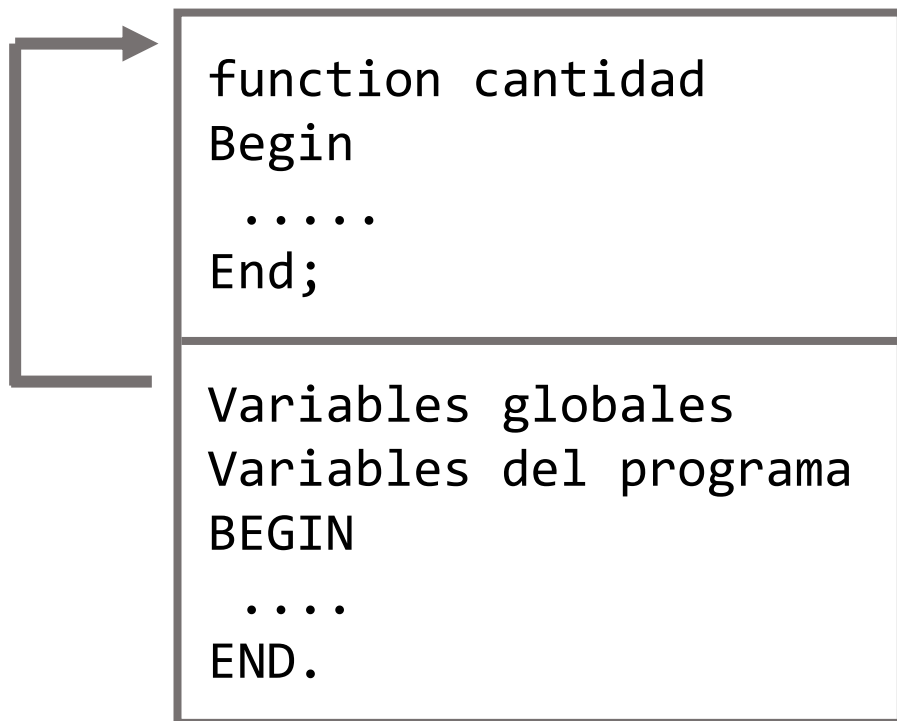




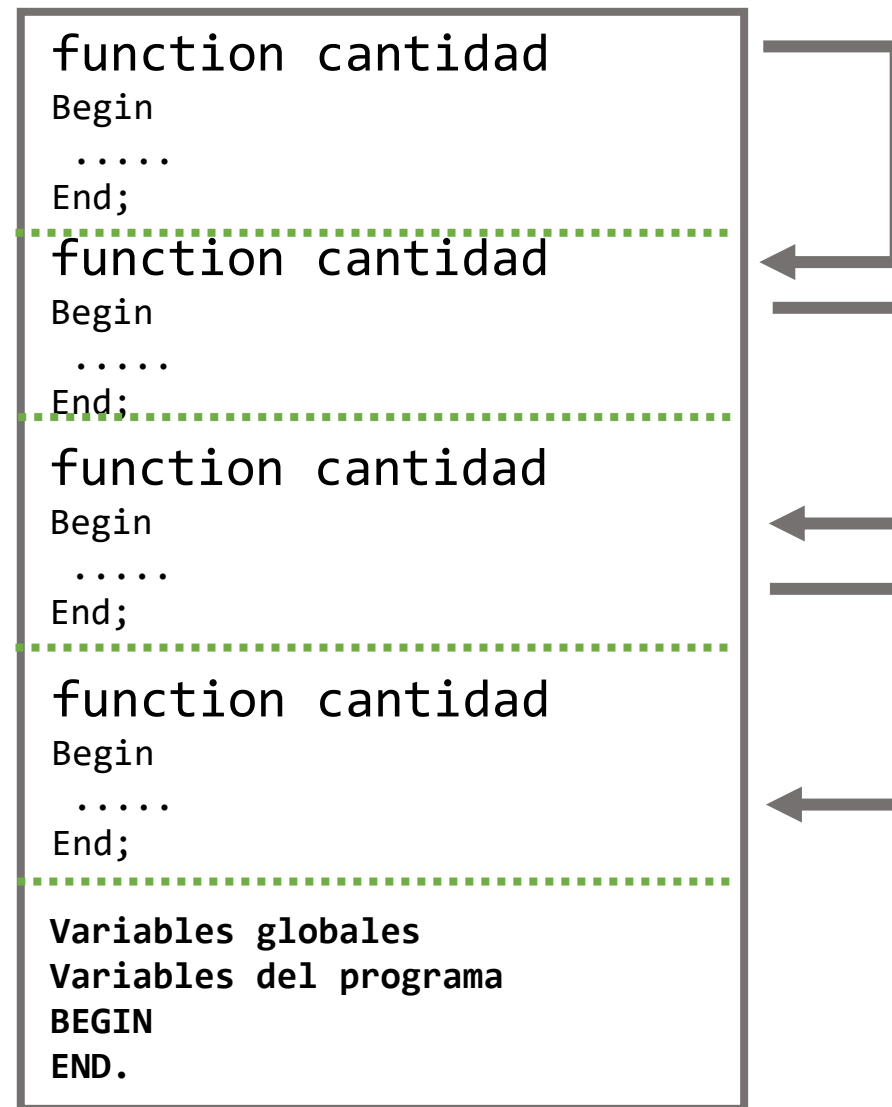
# Recursión - MOTIVACION



## ITERATIVA



## RECURSIVA





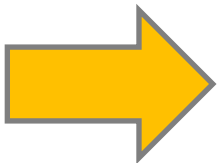
# RECURSION



Existen un conjunto de problemas que pueden resolverse siempre de la misma manera con la característica que el problema debe ir “achicandose” en cada instancia a resolver, hasta que en alguna instancia la solución es “trivial”.



La **recursividad** es una técnica de resolución de problemas que consiste en dividir un problema en instancias más pequeñas del mismo problema (también llamados subproblemas) hasta que obtengamos un subproblema lo suficientemente pequeño que tenga una solución trivial o directa.



La recursividad consiste en resolver un problema por medio de un módulo (procedimientos o funciones) que se llama a sí mismo, evitando el uso de bucles y otros iteradores.

Cuando el problema se va achicando llega a un punto que no puede achicarse más, esa instancia se denomina **caso base**.

Hay problemas en los cuales debe realizarse alguna tarea cuando se alcanza el caso base y otros que no. Hay problemas que pueden tener más de un caso base.