



Trabajo Práctico 6 Recursividad

Versión 7

IMPORTANTE: En RPA, para los enunciados donde se pide resolver el problema recursivamente, **no debe utilizar** ninguna estructura repetitiva de las vistas anteriormente (*for*, *while* o *repeat*). Además para cada ejercicio deberá implementarse un programa de prueba que llame a la/s primitiva/s implementadas.

Ejercicio 1. Considere el siguiente planteo recursivo que resuelve el problema de *calcular el dígito más significativo de un número entero*. Ejemplo: *digito_mas_significativo(4319)* retornará 4.

Planteo: **dígito más significativo de N**

Caso Base: si N tiene sólo un dígito entonces el dígito más significativo de N es N

Caso General: si N tiene más de un dígito entonces

el dígito más significativo de N es **el dígito más significativo de N sin su último dígito.**

Implemente en Pascal una **función** que respete el planteo propuesto para calcular el dígito más significativo.

Ejercicio 2. Escriba un planteo recursivo y luego implemente en Pascal una función que respete el planteo propuesto, para calcular la cantidad de dígitos de un número entero. Ejemplo: *cant_dig(45424)* retornará 5.

Ejercicio 3. Considere el siguiente planteo recursivo, que indica cuantos elementos de un archivo de enteros son positivos. Escriba una función en Pascal que respete este planteo y lo implemente.

Planteo: cantidad de positivos en el archivo A

Caso Base: si el archivo está vacío entonces la cantidad de positivos es 0

Caso General: si el archivo tiene elementos entonces

Si el primer elemento es positivo

entonces la cantidad de positivos es **la cantidad de positivos en el archivo A sin considerar su primer elemento** más 1

de lo contrario, la cantidad de positivos es **la cantidad de positivos en el archivo A sin considerar su primer elemento**

Ejercicio 4. Escriba un planteo recursivo y luego implemente en Pascal una función que respete el planteo propuesto, para determinar si un dígito D no pertenece a un número entero positivo N.

Ejemplo: *digito_ausente(5,1323)* retornará Verdadero, y *digito_ausente(1,1323)* retornará Falso.

Ejercicio 5. Escriba un planteo recursivo y luego implemente en Pascal una función que respete el planteo propuesto, para determinar si un dígito D está ubicado en la posición más significativa de un número natural.

Ejemplo: *pmasS (2,2345)* es verdadero, *pmasS (2,2325)* es verdadero, *pmasS (6,5604)* es falso y *pmasS (7,945)* es falso.



Ejercicio 6. Considere el siguiente planteo recursivo, que determina si todos los elementos de un archivo de entero son positivos. Escriba una función en Pascal que respete este planteo y lo implemente.

Planteo: **no hay negativos en el archivo A**

Caso Base: si el archivo está vacío entonces no hay negativos en el archivo A es VERDADERO

Caso General: si el archivo tiene elementos entonces

Si el primer elemento es negativo, entonces no hay negativos en el archivo A es FALSO

de lo contrario, no hay negativos en el archivo A si **no hay negativos en el archivo A sin considerar su primer elemento**

Ejercicio 7. Considere el siguiente planteo recursivo, extraído del parcial del año 2012, que resuelve el problema de calcular la cantidad de dígitos presentes en un número N que sean menores a un dígito D dado. Por ejemplo, *menores(123456,4)* retornará 3, *menores(8756,5)* retornará 0 y *menores(123123,4)* retornará 6. Implemente en Pascal una **función** que respete el planteo propuesto.

Planteo: **dígitos de N menores a D:**

Caso Base: si N tiene sólo un dígito entonces

si N es menor a D entonces la cantidad de dígitos de N menores a D es 1 sino es 0

Caso General:

si N tiene más de un dígito entonces

si el último dígito de N es menor a D

entonces la cantidad de dígitos de N menores a D es **la cantidad de dígitos de N sin su último dígito** menores a D más 1

sino la cantidad de dígitos de N menores a D es **la cantidad de dígitos de N sin su último dígito**

Ejercicio 8. Escriba un planteo recursivo e implemente en Pascal los siguientes incisos realizando procedimientos o funciones que respeten el planteo propuesto:

- Contar la cantidad de dígitos pares en un número entero. Ej.: *cantPares(22005)* el resultado es 4, *cantPares(35)* el resultado es 0, y *cantPares(484)* el resultado es 3.
- Determinar si un número natural P es prefijo de un número natural Q . Ej.: *esPrefijo(25,2545)* es verdadero, *esPrefijo(4,5604)* es falso, *esPrefijo(459,45)* es falso, *esPrefijo(25,25)* es verdadero.
- Dado un número entero N retorne el menor dígito presente en N . Por ejemplo *menor(123156)* retornará 1, *menor(8756)* retornará 5 y *menor(99999)* retornará 9.
- Determine si los dígitos de un número natural están dispuestos de forma creciente, es decir, si $N = d_m d_{m-1} \dots d_1 d_0$ y se verifica que $d_{i+1} \leq d_i \quad \forall i(0 \leq i \leq m)$. Por ejemplo: para 1227, 359, 88 o 139 debería retornar verdadero.



Ejercicio 9. Dada una secuencia ingresada por teclado de números enteros positivos finalizada en -1 (**el cual no se considera parte de la misma**), escribir un planteo recursivo y la correspondiente implementación (función o procedimiento) para:

- Mostrar los números pares de la secuencia en el mismo orden que fueron ingresados.
- Mostrar los números crecientes (ver ej 8 d) de la secuencia en orden inverso al que fueron ingresados.
- Sumar todos los enteros de dichas secuencia. *Ej.: Para la secuencia 2 5 3 6 12 3 -1 el resultado es 31.*

Ejercicio 10:

- Escriba un planteo recursivo y un **procedimiento** recursivo en Pascal (que respete el planteo) que muestre los todos elementos de un archivo de enteros según las siguientes restricciones:
 - En primer lugar se deberán mostrar por pantalla los enteros positivos en el orden en que aparecen en el archivo,
 - luego es mostrará por pantalla el caracter @,
 - y a continuación los enteros negativos pero en orden inverso al que están en el archivo.
 - El número 0 no debe mostrarse.
 - Si el archivo está vacío, solo mostrará el símbolo @.

*Por ejemplo, si el archivo tiene la secuencia: 12 -4 -5 0 3 99 0 6 -1
entonces el programa deberá mostrar: 12 3 99 6 @ -1 -5 -4*
- Escriba un programa que use y llame adecuadamente al procedimiento del inciso anterior.

Ejercicio 11. Realice una traza suponiendo que se produce la siguiente llamada al procedimiento recursivo `ex237(6)` y muestre la información que se imprimirá en pantalla como resultado de su ejecución.

```

procedure ex237(n:integer);
begin
    if (n>0) then
        begin
            writeln(n);
            ex237(n-2);
            ex237(n-3);
            writeln(n);
        end;
    end;
end;
    
```

Ejercicio 12. Considere el siguiente enunciado y el planteo recursivo que resuelve el problema. Implemente en Pascal un **procedimiento recursivo que respete el planteo** propuesto:

Dado un número natural $N = d_1 \dots d_{k-1} d_k$, y determine cuántos dígitos múltiplos de 3 posee en las posiciones impares (note que el dígito más significativo de N se considera la posición 1). Por ejemplo, el 8735 posee un dígito en estas condiciones mientras que 3690 posee dos y 23467 no posee ninguno.

Planteo: "Contar múltiplos en N"

Caso base: si N tiene un único dígito

entonces la cantidad es 1 si N es múltiplo de 3, o 0 si N no es múltiplo de 3.

Caso general: N tiene más de un dígito

Si el último dígito de N está en posición impar y además es múltiplo de 3

entonces la cantidad es 1 + **Contar múltiplos en N sin su último dígito**

de lo contrario, la cantidad es la misma que **Contar múltiplos en N sin su último dígito**



Ejercicio 13. Dado un número natural, definiremos como su *número balanceParImpar* al número que se obtiene de sumar sus dígitos impares y restar sus dígitos pares. *Por ej.: el número balanceParImpar de 318547 es 4 esto es, $\text{balanceParImpar}(318547) = \text{balanceParImpar}(31854)+7 = \text{balanceParImpar}(3185) - 4 + 7 = \dots$* Escriba el planteo recursivo e implemente en Pascal una función obtenga su *número balanceParImpar*.

Ejercicio 14: Escriba un planteo recursivo y una **primitiva** recursiva en Pascal (que respete el planteo) para cada uno de los siguientes incisos (en forma independiente). Considere que dispone de un archivo de texto, que por ejemplo, contiene refranes, he aquí un extracto:

1	El que mucho se despide, pocas ganas tiene de irse.
2	Sí por razones peleas, es que careces de ellas.
3	De los escarmentados nacen los avisados.

- Buscar cuantas veces está un carácter ingresado por el usuario en el archivo.
- Mostrar el archivo por pantalla. Cada línea debe aparecer en el mismo orden que está en el archivo y cada carácter dentro de cada línea también. Al finalizar de mostrar el archivo por pantalla deberá mostrar el siguiente mensaje '<Fin del archivo>'.
 c) Mostrar el archivo por pantalla en orden completamente inverso, es decir, las líneas en orden inverso y cada línea con sus caracteres en orden inverso también.

1	.sodasiva sol necan sodatnemracse sol eD
2	.salle ed secerac euq se ,saelep senozar rop iS
3	.esri ed eneit sanag sacop ,edipsed es ohcum euq IE

Ejercicio 15. Realice una traza del siguiente programa escrito en PASCAL.

```

program inverso;
var n,m:integer;

function exponente(n,e:integer):integer;
begin
  if (e = 0) then exponente := 1
  else exponente := exponente(n,e-1) * n;
end;

function cantDigitos(n:integer):integer;
var cant:integer;
begin
  if (n > -10) and (n < 10) then cant := 1
  else cant := 1 + cantDigitos(n div 10);
  cantDigitos := cant;
end;

procedure invertir(n:integer; var inv:integer);
var aux:integer;
begin
  if (n > -10) and (n < 10) then inv := n
  else
    begin
      invertir(n div 10, aux);
      inv := (n mod 10) * exponente(10,cantDigitos(n)-1) + aux;
    end;
end;

begin
  n := 1234;
  invertir(n,m);
  writeln('El inverso de ', n, ' es:', m);
end.

```



Ejercicio 16. Realice un planteo recursivo y luego implemente una función o procedimiento recursivo, que se corresponda con ese planteo, para cada caso:

- Leer una cadena de caracteres de longitud arbitraria finalizada en # y mostrar la cadena en orden inverso. *Ej.: si se tipea `animal#` deberá imprimirse en pantalla `lamina`*
- Leer una cadena de caracteres de longitud arbitraria finalizada en # y mostrar la cadena en orden inverso sin mostrar las vocales. *Ej.: si se tipea `animal#` deberá imprimirse en pantalla `lmm`*

Ejercicio 17. Realice procedimientos recursivos que procesen una secuencia de caracteres ingresada por teclado y finalizada en "." y muestren:

- La secuencia original (sin el terminador) seguida de secuencia original pero con sus caracteres en orden inverso. *Por ejemplo, si la secuencia es `A1#B2-C3$D.` deberá imprimir `A1#B2-C3DD3C-2B#1A`*
- Los caracteres de la secuencia original que se encuentran en posiciones impares en orden creciente, seguidos por los caracteres que se encuentran en posiciones pares en orden decreciente. Se considerará como posición 1 a la posición del primer caracter de la secuencia, posición 2 a la posición del segundo caracter y así sucesivamente. Por ejemplo, si la secuencia es `ABCDEF.` deberá imprimir `ACEFDB`

$$\begin{array}{cccccccc}
 A & B & C & D & E & F & . & \\
 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\
 \end{array}
 \rightarrow
 \begin{array}{cccccccc}
 A & C & E & F & D & B & & \\
 1 & 3 & 5 & 6 & 4 & 2 & &
 \end{array}$$

OBS: Tenga en cuenta que para resolver este ejercicio no debe usar archivos

Ejercicio 18. Para cada uno de los siguientes incisos, escriba un planteo recursivo y una función recursiva que respete el planteo realizado.

- Suma:** $N \times N \rightarrow N$ utilizando solamente como primitivas `succ` y `pred`. *Ej.: $\text{suma}(3,4) = \text{suma}(3,3) + 1 = \text{succ}(\text{suma}(3,\text{pred}(4)))$.*
- Resto:** $N \times N \rightarrow N$ que obtenga el resto (módulo) de la división entera utilizando como única operación aritmética la resta (no puede usarse `div`). *Ej.: $\text{resto}(5,2) = 1$, $\text{resto}(8,2) = 0$, $\text{resto}(1,2) = 1$.*
- DivEntera:** $N \times N \rightarrow N$ que obtenga el cociente (resultado) de la división entera utilizando como únicas operaciones aritméticas la suma y la resta.

Ejercicio 19: En este ejercicio se evaluará **recursión**. (Parcial 2015)

- Escriba un planteo recursivo y una **función** recursiva en Pascal (que respete el planteo) que dado un número entero determine si todos sus dígitos son impares. Ejemplo: para `N=311` retorna `TRUE` y para `101` retorna `FALSE`.
- Escriba un planteo recursivo y un **procedimiento** recursivo en Pascal (que respete el planteo) que dado un número entero positivo retorne la suma de los dígitos impares. *Ejemplo: si `N=341` retorna 4 y para `222` retorna 0.*
- Escriba un programa en Pascal que utilizando las primitivas de los incisos anteriores, permita leer un archivo `A` de enteros positivos llamado `naturales.nros` y generar otro archivo `cumple.nros` con todos los elementos del archivo `A` **que tengan todos sus dígitos impares, o bien que la suma de sus dígitos impares sea mayor a 10.**
- Indique un contenido posible del archivo `naturales.nros` para testear los casos posibles para el inciso c.

Ejercicio 20: (Recuperatorio 2015)

- Escriba un planteo recursivo y una función recursiva en Pascal (que respete el planteo) que dado un número `N` y un archivo de números enteros determine cuantos elementos del archivo son múltiplos de `N`.
- Escriba un planteo recursivo y un procedimiento recursivo en Pascal (que respete el planteo) que dado un número entero positivo retorne el producto de sus dígitos. *Ejemplo: `1234` retorna 24, y `1112` retorna 2, y `9083` retorna 0.*



- c) Escriba un programa en Pascal que utilizando las primitivas de los incisos anteriores, permita leer un archivo A de enteros positivos llamado `naturales.nros` y muestre por pantalla un mensaje indicando si la cantidad de múltiplos de 3 es mayor a la cantidad e múltiplos de 5, si es igual o si es menor. Además deberá generar otro archivo `cumple.nros` con todos los elementos del archivo A cuyo producto sea mayor a 10.
- d) Indique un contenido posible del archivo `naturales.nros` para testear los casos posibles para el inciso c.

Ejercicio 21: (Recuperatorio 2014)

- a) Escriba un planteo recursivo y una función recursiva en Pascal (que respete el planteo) que dado un archivo de texto cuente cuantas líneas tienen más caracteres que un determinado valor dado. *Por ejemplo, si el valor dado es 80 debería determinar cuantas líneas tienen más de 80 caracteres (mayor estricto).*

Escriba además un programa de prueba que solicite al usuario una cantidad y llame adecuadamente a dicha función con dicha cantidad, mostrando el resultado de la misma por pantalla.

- b) Escriba un planteo recursivo y un procedimiento recursivo en Pascal (que respete el planteo) que determine si un número natural es amistoso. Diremos que un número natural es amistoso si el dígito más significativo es divisible por 1, y el número formado por sus dos primeros dígitos es divisible por 2, y el número por sus tres primeros dígitos es divisible por 3, y así sucesivamente, hasta verificar que el número original es divisible por la cantidad de dígitos que lo componen. Por ejemplo, 42325 es amistoso, ya que 4 es divisible por 1, 42 es divisible por 2, 423 es divisible por 3, 4232 es divisible por 4 y 42325 es divisible por 5. Escriba además un programa de prueba que use y llame adecuadamente al procedimiento recursivo.

Ejercicio 22: (Parcial 2013)

- a) Escriba un planteo recursivo y una función recursiva en Pascal (que respete el planteo) que dado un número entero N retorne el menor dígito presente en N. Por ejemplo `menor(123156)` retornará 1, `menor(8756)` retornará 5 y `menor(99999)` retornará 9. Escriba además un programa de prueba para leer un número y mostrar el resultado de esta función en pantalla.
- b) Escriba un planteo recursivo y un procedimiento recursivo en Pascal (que respete el planteo) que muestre los todos elementos de un archivo de enteros según las siguientes restricciones:
- En primer lugar se deberán mostrar por pantalla los enteros positivos en el orden en que aparecen en el archivo,
 - luego es mostrará por pantalla el caracter @,
 - y a continuación los enteros negativos pero en orden inverso al que están en el archivo.
 - El número 0 no debe mostrarse.
 - Si el archivo está vacío, solo mostrará el símbolo @.
- Por ejemplo, si el archivo tiene la secuencia: **12 -4 -5 0 3 99 0 6 -1**
entonces el programa deberá mostrar: **12 3 99 6 @ -1 -5 -4**
- c) Escriba un programa que use y llame adecuadamente al procedimiento del inciso anterior.

Ejercicio 23: (Recuperatorio 2014)

- a) Escriba un planteo recursivo y una función recursiva en Pascal (que respete el planteo) que dado un número natural determine si el número contiene algún par de dígitos iguales en forma consecutiva. Por ejemplo, `consecutivos(1234)` retornará falso, `consecutivos(12334)` retornará verdadero, `consecutivos(3234)` retornará falso y `consecutivos(4423211)` retornará verdadero.
- b) Escriba un planteo recursivo y un procedimiento recursivo en Pascal (que respete el planteo) que dado un archivo de números reales que representa todos los gastos realizados en un viaje, retorne el monto total gastado (y si el archivo está vacío cero); y además muestre por pantalla, en orden inverso al que se encuentran en el archivo, los montos superiores a un valor ingresado por el usuario.
- c) Realice un programa que solicite al usuario un valor y llame adecuadamente al procedimiento implementado en el inciso (b) (no es necesario volver a escribir el procedimiento).