



Trabajo Práctico N° 9 Recursividad

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*).
- El planteo es parte fundamental de la forma de resolución de un problema recursivo.
- 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*.

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**.

Ejemplo: `digito_mas_significativo(4319)` retornará 4.

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 recursiva que se corresponda con esa definición, para calcular la **división entera entre dos números A y B**, utilizando como operaciones aritméticas la suma y la resta.

Ejercicio 3.

Escriba un planteo recursivo y luego implemente en Pascal una función recursiva que se corresponda con ese planteo, para que dado un entero **N**:

- a) Calcule la **cantidad de dígitos** de N.
- b) Calcule el **mayor dígito** de N.

Ejercicio 4.

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

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 no está vacío, 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 5.

Escriba un planteo recursivo y luego implemente en Pascal una función recursiva que se corresponda con ese planteo, para que dado un entero **N** y un dígito **d**:

- a) Calcule la cantidad de apariciones del dígito d en N.
- b) Decida si d es el dígito más significativo de N.
- c) Decida si d está ausente en N.



Ejercicio 6.

Escriba el planteo recursivo e implemente un procedimiento o función, que se corresponda con ese planteo, que determine si los dígitos de un número natural N están dispuestos de forma **creciente**, esto es, si $N = d_m d_{m-1} \dots d_1 d_0$ y $\forall i (0 \leq i \leq m)$ se verifica que $d_{i+1} \leq d_i$.

Por ejemplo: para 1227, 359, 88 o 139 debería retornar verdadero, y para 1221, falso.

Ejercicio 7.

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.

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 8.

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.

MENORES:

Caso Base: si N tiene sólo un dígito entonces, si N es menor a D , **MENORES** 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 , **MENORES** de N es **MENORES** de N sin su último dígito más 1, sino **MENORES** de N es **MENORES** de N sin su último dígito.

Ejercicio 9.

Dado un número natural, definiremos como su **número promedio** al número que se obtiene de sumar sus dígitos impares y restar sus dígitos pares.

Por ejemplo, el número promedio de 1227 es $1 - 2 - 2 + 7 = 4$

Escriba el planteo recursivo e implemente en Pascal una función para obtener el número promedio de N .

Ejercicio 10.

Escriba un planteo recursivo y luego implemente en Pascal una función recursiva para que dado un número natural encuentre el número AIMPAN que resulta de reemplazar cada dígito **par** por su sucesor.

Por ejemplo, si $N=11245$, el número AIMPAN es 11355.

$N=1218$, el número AIMPAN ES 1319.

Ejercicio 11.

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.

Ejercicio 12.

Dada una secuencia de caracteres terminada en #, escriba un planteo recursivo y luego implemente en Pascal un procedimiento recursivo, que se corresponda con ese planteo, para:

- Mostrarla en **orden inverso**.
- Mostrarla en **orden inverso sin mostrar las vocales**.



Resolución de Problemas y Algoritmos

Profesor: Luciano Tamargo

Asistente: Natalia Nill



Ejercicio 13.

Dada una secuencia de números terminada en -1 (el cual no pertenece a la secuencia), escriba un planteo recursivo y luego implemente en Pascal una función recursiva, que se corresponda con ese planteo, para

- Contar la **cantidad de números** de la secuencia.
- Sumar todos los números de la secuencia.
- Calcular el **mayor** número de la secuencia.
- Decidir si todos son números **positivos**.

Ejercicio 14.

Dado un número m y una secuencia de números terminada en -1 (el cual no pertenece a la secuencia), escriba un planteo recursivo y luego implemente en Pascal una función recursiva, que se corresponda con ese planteo, para

- Calcular cuántos números son múltiplos de m .
- Decidir si existe algún número primo mayor a m .
- Contar la cantidad de apariciones del número m .
- Decidir si m está ausente en la secuencia.

Ejercicio 15.

a) Escriba un planteo recursivo y un procedimiento recursivo en Pascal (que respete el planteo) que muestre todos los elementos de un archivo de enteros según las siguientes restricciones:

- Mostrar por pantalla los enteros positivos en el orden en que aparecen en el archivo,
- luego mostrar por pantalla el caracter @,
- y a continuación los enteros negativos pero en orden inverso al que están en el archivo.
- Si el archivo está vacío, solo mostrará el símbolo @.

Por ejemplo:

Si el archivo tiene la secuencia: 12 -4 -5 3 99 0 6 -1 entonces el programa deberá mostrar: 12 3 99 6 @ -1 -5 -4

b) Escriba un programa que use y llame adecuadamente al procedimiento del inciso anterior.

Ejercicio 16.

Considere una secuencia de números ingresada por teclado que consiste de una cantidad de $(2k+1)$ valores para $k > 0$. Se desconoce el valor de k , pero se conoce que la secuencia contiene en la posición $(2k+1)/2$, es decir, en la mitad de la secuencia el número -1 y dicho número no es utilizado en ninguna otra posición.

Por ejemplo, una posible secuencia es: 2 8 7 -25 3 -1 44 10 -5 6 1.

Implemente un procedimiento recursivo que muestre los números de la secuencia **apareados** de la siguiente forma: el primer número de la secuencia apareado con el último, el segundo con el anteúltimo y así hasta alcanzar al valor -1 el cual es descartado. Se deberá mostrar los pares entre paréntesis. Por ejemplo, si consideramos la secuencia anterior se deberían mostrar los pares (3,44) (-25,10) (7,-5) (8,6) (2,1).

Ejercicio 17. (Recuperatorio 2014)

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.

Ejercicio 18.

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 19. 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 cantDígitos(n:integer):integer;
  var cant:integer;
  begin
    if (n > -10) and (n < 10) then cant := 1
    else cant := 1 + cantDígitos(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,cantDígitos(n)-1) + aux;
      end;
  end;
begin
  n := 1234;
  invertir(n,m);
  writeln('El inverso de ', n, 'es:', m);
end.

```

Ejercicio 20.

Una secuencia $S = a_1 a_2 \dots a_k b_k \dots b_2 b_1$ de **longitud par** se dice que “**palíndromo**” si para todo i en $[1..k]$ se verifica $a_i = b_i$.

Por ejemplo: $S = abcd\bar{b}b$ (longitud=6) **no es palíndromo**, $S = abba$ (longitud=4) **es palíndromo**, Si S es una secuencia vacía (longitud=0) **es palíndromo**.

Dada esta definición, proponga un planteo recursivo e implemente una función para decidir si una secuencia de caracteres es capicúa.