



Trabajo Práctico N° 10

Recursividad

Ejercicio 1. Implemente en Pascal las siguientes definiciones recursivas.

$$a) \quad h(N) = \begin{cases} 1 & , N = 0 \\ h(N-1) + h(N-1) & , N > 0 \end{cases}$$

$$b) \quad g(x, y) = \begin{cases} 0 & , x = 0 \\ 0 & , y = 0 \\ 1 + g(x-1, y-1) & , x > 0 \wedge y > 0 \end{cases}$$

$$c) \quad f(i) = \begin{cases} 1 & , i = 0 \\ 1 + f(i-1) & , \text{si } i \text{ es par} \\ f(i-1) & , \text{si } i \text{ es impar} \end{cases}$$

$$d) \quad knuth(i) = \begin{cases} 1 & , i = 0 \\ 1 + \min(2 * knuth(\lfloor i/2 \rfloor), 3 * knuth(\lfloor i/3 \rfloor)) & , i \geq 1 \end{cases}$$

Ejercicio 2. Considere el siguiente programa escrito en PASCAL y realice una traza suponiendo que se ingresa 1234:

```
program inverso;

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

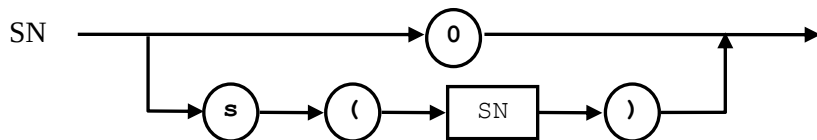
procedure invertir(n:integer; var inv:integer);
var aux:integer;
begin
    if (n > -10) and (n < 10) then inv := n
    else
        begin
            invertir(n div 10, aux);
            inv := (n mod 10) * exponente(10,cantDigitos(n)-1) + aux;
        end;
end;

var n,m:integer;
begin
    write('Ingrese un nro:');
    readln(n);
    invertir(n,m);
    writeln('Su inverso es:',m);
end.
```



Ejercicio 3. Los números naturales pueden ser representados como una función unaria usualmente llamada s^n .

Por ejemplo:



$0 = 0$
 $1 = s(0)$
 $2 = s(s(0))$
 $3 = s(s(s(0)))$
 $7 = s(s(s(s(s(s(s(0)))))))$

- Realice un procedimiento recursivo que reciba un entero positivo y muestre por pantalla la representación de dicho número en formato $s^n(0)$.
- Realice una función recursiva que reciba una secuencia de caracteres ingresada por teclado representando un número en notación $s^n(0)$ retorne el entero correspondiente.

Ejercicio 4. Escriba el planteo recursivo e implemente en Pascal los siguientes incisos teniendo en cuenta las restricciones impuestas para cada caso. Obviamente, podrán ser utilizadas en todos los casos las estructuras de control y llamadas a procedimientos y funciones necesarios siempre y cuando los mismos también respeten las restricciones antes mencionadas.

- Una función recursiva **suma**: $N \times N \rightarrow N$ utilizando solamente como primitivas `succ` y `pred`.
- Una función recursiva **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.: $\text{resto}(5,2) = 1$, $\text{resto}(8,2) = 0$, $\text{resto}(1,2) = 1$.
- Una función recursiva **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.
- Una función recursiva **cuadrado**: $N \rightarrow N$ que obtenga el cuadrado de un número natural distinto de cero utilizando exclusivamente el siguiente método: el $\text{cuadrado}(k)$ es igual a la suma de los k primeros números impares. Por ejemplo, el cuadrado de 4 es $1+3+5+7=16$.

Ejercicio 5. Escriba un planteo recursivo e implemente ambos incisos en Pascal considerando cuidadosamente si los definirá como procedimientos o como funciones:

- Mostrar los números del 1 al N en orden creciente.
- Mostrar los números del 1 al N en orden decreciente.

Ejercicio 6. Escriba un planteo recursivo e implemente en Pascal los siguientes incisos:

- Una **función** recursiva que determine si un dígito D no pertenece a un número entero positivo N . Ej.: si $N=1323$ y $D=5$ el resultado es Verdadero, y si $D=1$ el resultado es Falso.
- Una función o procedimiento que cuente la cantidad de dígitos pares en un número entero. Ej.: si el número es 22005 el resultado es 4, y si fuera 35 el resultado es 0.
- Una función o procedimiento que cuente la cantidad de dígitos pares que ocupan posiciones impares (identificándolas de izquierda a derecha) en un número entero. Ej.: si el número es 22005 el resultado es 2, y si fuera 1414 el resultado es 0



Ejercicio 7. Considere la siguiente función recursiva:

```
function misterio(a,b:integer):integer;
begin
    if (b = 0) then misterio := 0
    else if (b mod 2 = 0) then misterio := misterio(a+a, b div 2)
        else misterio := misterio(a+a, b div 2) + a;
    end;
```

a) Realice una traza para las siguientes llamadas:

- i. `misterio(2,25);`
- ii. `misterio(3,11);`

b) Determine que función matemática define `misterio`.

c) Si se reemplaza en la línea 3 por `misterio := 1` y las operaciones `+` por `*` en las líneas 4 y 5, ¿Qué función matemática queda definida?

Ejercicio 8. Escriba un planteo recursivo e implemente en Pascal los siguientes incisos:

- a) Una **función** recursiva que determine si un número natural es potencia de 2. Ej.: `espot2(33)=false`, `espot2(64)=true`.
- b) Una **función** recursiva que determine si dígito D está ubicado en la posición más significativa de un número natural. Ej.: `pmasS(2,2345) = true`, `pmasS(6,5604) = false`, `pmasS(7,945) = false`.
- c) Una **función** recursiva que determine si un número natural P es prefijo de un número natural Q. Ej.: `esPrefijo(25,2545)= true`, `esPrefijo(4,5604)= false`, `esPrefijo(459,45)= false`, `esPrefijo(25,25)= true`.
- d) Una **función** recursiva que determine si un número natural P es sufijo de un número natural Q. Ej.: `esSufijo(25,2545)= false`, `esSufijo(4,5604)= true`, `esSufijo(459,45)= false`, `esSufijo(25,25)= true`.

Ejercicio 9. 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;
```

Ejercicio 10. Escriba un planteo recursivo y defina procedimientos o funciones recursivos para cada caso:

- a) 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`
- b) 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 `lmn`



Ejercicio 11. Suponiendo que cuenta con un archivo de caracteres F , escriba un planteo recursivo y defina procedimientos o funciones para cada caso:

- c) Mostrar el contenido del archivo en orden inverso pero las mayúsculas como minúsculas y las minúsculas como mayúsculas.

*Ej.: si el archivo tiene los caracteres **AniMal feRoz** deberá imprimirse en pantalla **ZOrEF LAmINa***

- d) Mostrar el contenido del archivo de la siguiente manera: los dígitos en el orden ingresado, luego las letras en orden inverso, considerando que los demás caracteres no deben imprimirse.

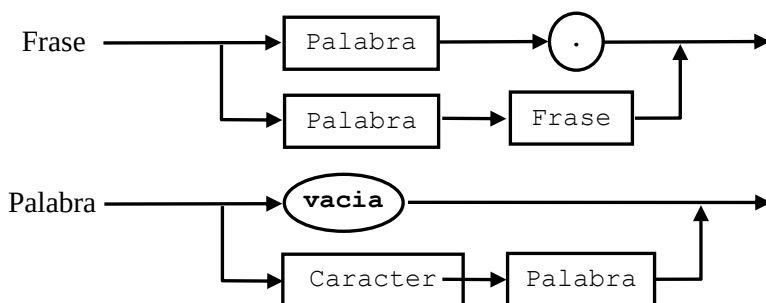
*Ej.: si el archivo tiene los caracteres **12 ani 4 + mal 6** deberá imprimirse en pantalla **1246lamina***

Ejercicio 12. Realice procedimientos recursivos que procesen una secuencia de caracteres finalizada en "." y muestren:

- a) 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-C3\$DD\$3C-2B#1A***
- b) 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 carácter de la secuencia, posición 2 a la posición del segundo carácter y así sucesivamente. *Por ejemplo, si la secuencia es **ABCDEF.** deberá imprimir **ACEFDB***

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

Ejercicio 13. Defina procedimientos o funciones recursivos que permita leer una frase terminada en punto y contar la cantidad de palabras de dicha frase. *Por ejemplo, si la frase ingresada es "Que lindo día." deberá devolver 3, y si la frase fuera "." deberá devolver 0.*



Ejercicio 14. Escriba un planteo recursivo y defina procedimientos o funciones recursivos para cada caso:

- a) Leer un archivo de caracteres y determinar si se encuentra ordenado alfabéticamente.
- b) Leer un archivo de caracteres y determinar la cantidad de vocales y consonantes del mismo
- c) Leer un archivo de caracteres y determinar si hay algún diptongo en el mismo.

Ejercicio 15. Escriba el planteo recursivo e implemente en Pascal una función recursiva que calcule la suma de los dígitos que ocupan posiciones impares para un número natural. Se considera que la posición 1 es la posición del dígito menos significativo (lugar de la unidad), la posición 2 es la posición de la decena, etc. *Por ejemplo, si se considera el natural 587, el 7 está en la posición 1, el 8 en la posición 2 y el 5 en la posición 3. En el ejemplo, la función debería retornar 12 (7+5).*

Ejercicio 16. Escriba el planteo recursivo e implemente en Pascal un procedimiento o función que determine si los dígitos de un número natural 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.*



Ejercicio 17. 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 ej.: el número promedio de 318547 es 4 esto es, $\text{numeroPromedio}(318547) = \text{numeroPromedio}(31854) + 7 = \text{numeroPromedio}(3185) - 4 + 7 = \dots$*

Escriba el planteo recursivo e implemente en Pascal una función obtenga su **número promedio**.

Ejercicio 18. Escriba un planteo y un procedimiento recursivo para imprimir una media pirámide de dígitos como se muestra en la siguiente figura. Utilice un procedimiento recursivo para generar cada fila de la media pirámide.

```

1
21
321
4321
54321
654321
7654321
87654321
987654321

```

Ejercicio 19. Teniendo en cuenta las siguientes definiciones recursivas para la función potencia:

$$x^n = \begin{cases} 1 & , n = 0 \\ x * x^{n-1} & , n > 0 \end{cases}$$

Potencia1

$$x^n = \begin{cases} (x^2)^{n/2} & \text{si } n \text{ es par} \\ x * (x^2)^{(n-1)/2} & \text{si } n \text{ es impar} \end{cases}$$

Potencia2

- Realice funciones para cada una de las definiciones.
- ¿Cuántas llamadas a la función potencia se requerirán para computar 3^6 , 3^7 y 3^8 ?
- Compare cuántas llamadas recursivas se realizan en potencia1 y en potencia2 para los casos mencionados en el inciso anterior.

Ejercicio 20. Dada una secuencia 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:

- Sumar todos los enteros de dichas secuencia. *Ej.: Para la secuencia 2 5 3 6 12 3 -1 el resultado es 31.*
- Mostrar por pantalla todos los valores de la secuencia que sean divisibles por el último valor de la misma.
- Calcular el promedio de los valores de la secuencia.
- Determinar el k-esimo elemento de la secuencia comenzando desde adelante. El valor k debe ser proporcionado por el usuario. *Ej.: Para la secuencia 2 5 3 6 12 3 -1 y $k = 4$ el resultado es 6.*

Ejercicio 21. Resuelva los **incisos a y b** del ejercicio anterior, pero considerando que cuenta con un archivo de números enteros