



Trabajo Práctico N° 9 Recursividad

Ejercicio 1. Implemente en Pascal las siguientes definiciones recursivas.

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

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

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

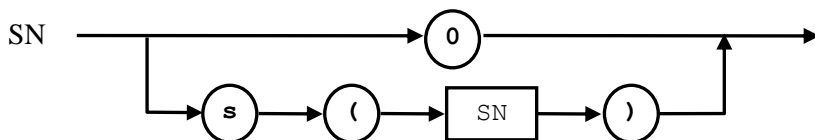
Ejercicio 2.

- Escriba un planteo recursivo y luego implemente en Pascal una función recursiva, que se corresponda con ese planteo, para calcular el dígito más significativo de un número entero. Ejemplo: `mas_significativo(4349)` retornará 4.
- Escriba un planteo recursivo y luego implemente en Pascal una función recursiva, que se corresponda con ese planteo, para calcular la cantidad de dígitos de un entero. Ejemplo: `cant_dig(45424)` retornará 5.

Ejercicio 3. 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`.
- 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.: `resto(5,2) = 1`, `resto(8,2) = 0`, `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 4. Los números naturales pueden ser representados como una función unaria usualmente llamada s^n .



Por ejemplo:

$$\begin{aligned} 0 &= 0 \\ 1 &= s(0) \\ 2 &= s(s(0)) \\ 3 &= s(s(s(0))) \\ 7 &= s(s(s(s(s(s(s(0))))))) \end{aligned}$$

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



Ejercicio 5. Considere el siguiente programa escrito en PASCAL y realice una traza suponiendo que se ingresa 1234

```
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;
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;

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

Ejercicio 6. Escriba un planteo recursivo e implemente en Pascal los siguientes incisos realizando primitivas que respeten el planteo propuesto:

- Realice un planteo recursivo y luego implemente una función recursiva llamada “digito_ausente”, que se corresponda con ese planteo, que determine si un dígito D no pertenece a un número entero positivo N. Ej.: si $N=1323$ y $D=5$ el resultado será Verdadero, y si $D=1$ el resultado será Falso.
- Realice un planteo recursivo y luego implemente una función o procedimiento recursivo, que se corresponda con ese planteo, 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.
- Realice un planteo recursivo y luego implemente una función recursiva, que se corresponda con ese planteo, que determine si dígito D está ubicado en la posición más significativa de un número natural. Ej.: $pmasS(2,2345) = true$, $pmasS(6,5604) = false$, $pmasS(7,945) = false$.
- Realice un planteo recursivo y luego implemente una función recursiva, que se corresponda con ese planteo, que determine si un número natural P es prefijo de un número natural Q. Ej.: $esPrefijo(25,2545) = true$, $esPrefijo(4,5604) = false$, $esPrefijo(459,45) = false$, $esPrefijo(25,25) = true$.
- Realice un planteo recursivo y luego implemente una función recursiva, que se corresponda con ese planteo, que obtenga el mayor dígito de un número natural. Ejemplos: $mayorDigito(244) = 4$; $mayorDigito(4723) = 7$



Ejercicio 7. 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 8. 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 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 9. 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 10. Suponiendo que cuenta con un archivo de caracteres F, escriba un planteo recursivo y defina procedimientos o funciones para cada caso:

- 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 `ArroZ AniMal` deberá imprimirse en pantalla `LAmINa ZorrA`
- 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 11. Una frase se considera palíndromo si invertida sin considerar los espacios es igual a la original sin considerar los espacios y sin tener en cuenta acentos. Los siguientes son ejemplos de frases palíndromas: "`dábaLe arroz a La zorra el abad`"; "`a ti no bonita`"; "`ánimo Romina`"; "`La ruta nos aportó otro paso natural`"; "`Sé verLa al revés`". Utilizando recursión escriba un programa que determine si una frase contenida en un archivo de caracteres es o no un palíndromo. Asuma que cuenta con la función

`LetrasIguales(L1, L2:char):boolean`; ya implementada en Pascal, que retorna verdadero si dos letras dadas son iguales, independientemente de mayúsculas o acentos; por ejemplo `LetrasIguales('á', 'A')` retorna TRUE.



Ejercicio 12. 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-C3\$DD\$3C-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 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*

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

OBS: Tenga en cuenta que para resolver este problema, no se debe usar archivos

Ejercicio 13. 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, numeroPromedio(318547) = numeroPromedio(31854)+7 = numeroPromedio(3185) - 4 + 7 = ...*

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

Ejercicio 14. 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 15. 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 opcionales (para practicar para recuperatorios, finales, etc):

Ejercicio I. Implemente en Pascal las siguientes definiciones recursivas.

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

Ejercicio II. Escriba un planteo recursivo **cuadrado**: $N \rightarrow N$ que obtenga el cuadrado de un número natural distinto de cero utilizando exclusivamente el siguiente método: el 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.* Implemente una función en Pascal que respete el planteo dado.

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

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

Ejercicio IV. 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;
```

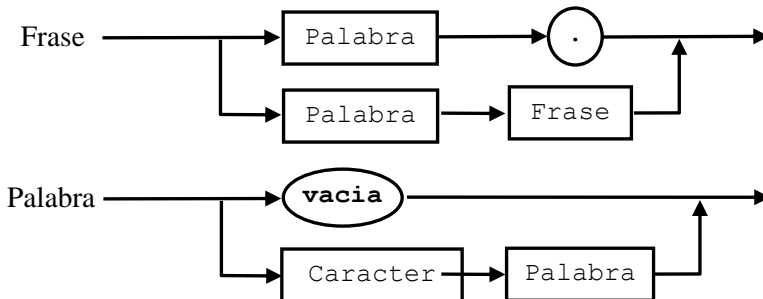
- a) Realice una traza para las siguientes llamadas:
 - i. misterio(2,25);
 - ii. misterio(3,11);
- b) Determine que función matemática define misterio.
- c) Si se reemplaza en la línea 3 por misterio := 1 y las operaciones + por * en las líneas 4 y 5, ¿Qué función matemática queda definida?

Ejercicio V. Escriba un planteo recursivo e implemente en Pascal

- a) 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*
- b) 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.*
- c) Una **función** recursiva que determine si un número natural es potencia de 2. *Ej.: espot2(33)=false, espot2(64)=true.*



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

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

Ejercicio VIII. 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 IX. 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.

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