Trabajo Práctico 7
Recursividad y Archivos de Texto

Para los ejercicios que solicita resolver el problema recursivamente no puede utilizar, en esta materia, ninguna estructura repetitiva de las vistas anteriormente (for, while o repeat)

Para los ejercicios que se encuentren identificados con el símbolo  *se deberá realizar un programa que declare un archivo de texto y los procedimientos o funciones que se solicitan.*

Ejercicio 1. Considere el siguiente planteo recursivo que resuelve el problema de *calcular el dígito más significativo de un número entero. Ejemplo: mas_significativo(4349) retornará 4.*

Planteo mas_significativo(N):

> Caso Base: si N tiene sólo un dígito entonces mas_significativo(N) es N
>
> Caso General: si N tiene más de un dígito entonces
>
> Sea M el número resultante de sacarle a N su último dígito
>
> luego mas_significativo(N) es mas_significativo(M)

implemente en Pascal una **función** que respete el planteo propuesto.

Ejercicio 2. 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 E 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.*

Planteo menores(N,D):

> Caso Base: si N tiene sólo un dígito entonces
>
> si N < D entonces menores(N,D) es 1 sino 0
>
> Caso General: si N tiene más de un dígito entonces
>
> Sea Cant la cantidad de dígitos de M menores a D (ie, menores(M,D)) donde M es el número resultante de sacarle a N su último dígito
>
> Si el último dígito de N es menor a D
>
> entonces Menores(N,D) es cant + 1
>
> sino Menores(N,D) es cant

implemente en Pascal una **función** que respete el planteo propuesto.

Ejercicio 3. Escriba un planteo recursivo e implemente en Pascal los siguientes incisos realizando procedimientos o funciones que respeten el planteo propuesto:

a) Calcular la cantidad de dígitos de un entero. *Ejemplo: cant_dig(45424) retornará 5.*

b) Determinar si un dígito D no pertenece a un número entero positivo N. *Ejemplo: dígito_ausente(5,1323) retornará Verdadero, y dígito_ausente(1,1323) retornará Falso.*

c) Contar la cantidad de dígitos pares en un número entero. *Ej.: cantPares(22005) el resultado es 4, y cantPares(35) el resultado es 0.*

d) Determinar si un dígito D está ubicado en la posición más significativa de un número natural. *Ejemplo: pmasS (2,2345) es verdadero, pmasS (6,5604) es falso y pmasS (7,945) es falso.*

e) 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.*

f) Obtenga el mayor dígito de un número natural. Ejemplos: *mayorDígito*(244) es 4; *mayorDígito*(4723) es 7

Ejercicio 4. Para cada uno de los siguientes incisos, escriba un planteo recursivo y una función recursiva que respete el planteo realizado.

a. ***Suma***: N×N→N utilizando solamente como primitivas `succ` y `pred`.

b. ***Resto***: N×N→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.*

c. ***DivEntera***: N×N→N que obtenga el cociente (resultado) de la división entera utilizando como únicas operaciones aritméticas la suma y la resta.

Ejercicio 5:

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 6: Preguntas Teóricas

a) ¿Qué es recursión? ¿Cuándo un algoritmo es recursivo? Muestre ejemplos de algoritmos recursivos válidos y no válidos (indicando porque no lo son).
b) Muestre diferencias y similitudes entre un archivo declarado de tipo predefinido TEXT y uno declarado como FILE OF CHAR.
c) Explique las diferencias y similitudes entre procedimientos y funciones en Pascal.
d) Explique cuando un identificador es visible en un bloque. Ejemplifique en Pascal.
e) Indique la definición de entorno de referencia de un bloque de Pascal.
f) Explique las diferencias entre parámetros por valor y por referencia. Indique además que es lo que se permite para cada uno de ellos en los parámetros efectivos.
g) Explique cuando dos tipos son idénticos, cuando compatibles y cuando se cumple la compatibilidad de asignación. Escriba 2 ejemplos de cada uno.
h) Explique por qué en Pascal se requiere declarar un identificador de tipo, cuando un procedimiento o una función tienen un archivo como parámetro.

Ejercicio 7: Escriba una función que busque cuantas veces está el caracter E (ingresado por el usuario) en un archivo de texto.

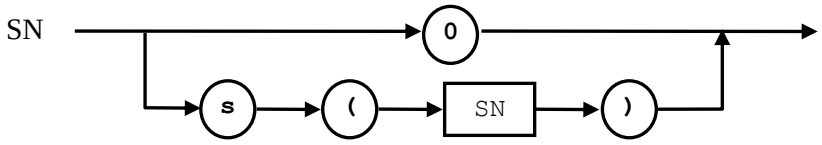
Ejercicio 8: Escriba un procedimiento que indique cuantas líneas tiene un archivo de texto.

Ejercicio 9: Escriba una función que busque cuantas veces está el caracter E (ingresado por el usuario) en cada línea de un archivo de texto.

Ejercicio 10: Escriba un procedimiento que agregue dos líneas en blanco a un archivo de texto. ¿Podría utilizarse una función para dicha tarea?

Ejercicio 11: Escriba un procedimiento o función que reemplace el carácter E por otro C (ingresados por el usuario) en un archivo de texto, indique en cuantas líneas lo reemplazó.

Ejercicio 12. 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)))))))

a. 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.

b. 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 retorne el entero correspondiente.

Ejercicio 13. 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 14. 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 $N = d_m d_{m-1} \ldots d_1 d_0$ y $\forall i(0 \le i \le m)$ se verifica que $d_{i+1} \le d_i$. *Por ejemplo: para 1227, 359, 88 o 139 debería retornar verdadero.*

Ejercicio 15. 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 cantDígitos(n:integer):integer;
        begin
             if (n > -10) and (n < 10) then cantDígitos := 1
             else cantDígitos := 1 + cantDígitos(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,cantDígitos(n)-1) + aux;
                 end;
        end;
begin
     write('Ingrese un nro:');
     readln(n);
     invertir(n,m);
     writeln('Su inverso es:',m);
end.
```

Ejercicio 16. Considere el siguiente enunciado y el planteo recursivo que resuelve el problema. Implemente en Pascal un **procedimiento recursivo que respete el planteo** propuesto:

Problema: Escriba un planteo recursivo y un procedimiento recursivo (que respete el planteo) que reciba un número natural $N = d_1 \ldots d_{k-1} d_k$, 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 17. Realice un planteo recursivo y luego implemente una función o procedimiento recursivo, que se corresponda con ese planteo, 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 18. Suponiendo que cuenta con un archivo de caracteres F, escriba un planteo recursivo y defina procedimientos o funciones para cada caso:

a) 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`**

b) 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 19. Realice procedimientos recursivos que procesen una secuencia de caracteres ingresada por teclado y 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 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`

$$\underset{1\ 2\ 3\ 4\ 5\ 6\ 7}{ABCDEF.} \rightarrow \underset{1\ 3\ 5\ 6\ 4\ 2}{ACEFDB}$$

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

Ejercicio 20. 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 21: Escriba una función que busque cuantas veces está una palabra de 3 letras (ingresado por el usuario) en un archivo de texto.

Ejercicio 22: Escriba una función que busque cuantas veces está el caracter 'A' seguido del carácter 'C' en un archivo de texto.

Ejercicio 23: Escriba un programa que lea un archivo de texto con pares LU Nota (donde Nota es A B C o D) y genere un nuevo archivo de texto que tenga un listado para imprimir con un alumno por línea con el formato LU <resultado> (EOL) . donde <resultado> es Aprobado si Nota es A o B y Desaprobado si es C o D.

Ejercicio 24: Suponga que dispone de un archivo de texto llamado "`impresora.conf`" que contiene las preferencias de una impresora. Por ejemplo, el contenido del archivo podría ser el siguiente

```
nombre impresora = LASER P-PITA
tamaño de hoja = A4
calidad = borrador
orientación = apaisada
predeterminada = si
```

En este ejemplo, cada línea corresponde a un par (preferencia, valor) donde el nombre de la preferencia se encuentra a la izquierda del signo igual y el valor se encuentra a su derecha.
Escriba un programa que muestre el contenido del archivo, numerando cada una de las preferencias y permita seleccionar al usuario (mediante un número) la preferencia a modificar. Luego de seleccionar la preferencia, se debe permitir que el usuario ingrese el nuevo valor y este deberá reflejarse modificando el archivo. Deberán existir dos opciones adicionales, cero en el caso de no querer modificar nada y menos uno (-1) si se desea agregar una nueva preferencia (no es necesario verificar que no sea un duplicado).

Ejercicio 25: *En este ejercicio se evaluará la correcta* ***división del problema en subproblemas****.*

Considere que una empresa de transportes de latas de bebidas tiene dos camiones y en cada uno de ellos puede llevar hasta 100 cajas de latas. Cada caja tiene un código individual único representado por un número entero el cual se utiliza para su seguimiento durante el transporte. La empresa tiene dos archivos de enteros "camión1" y "camión2", cada archivo con los códigos de las cajas que ya han sido cargadas en cada camión y que aún no ha salido de viaje. Además, para el nuevo lote de cajas que no fue cargado aún, la empresa tiene otro archivo de enteros llamado "lote_a_cargar" con los códigos de todas las cajas de ese lote.

Se debe escribir un programa en Pascal que leyendo la información de los archivos indicados antes, permita al operador del sistema realizar estas tareas: (1) mostrar cuántas cajas pueden aún cargarse en cada camión, (2) mostrar si dado el espacio disponible entre los dos camiones, alcanza para cargar todas las cajas de "lote_a_cargar", o de lo contrario cuantas cajas quedarían sin cargar, y (3) dado un código, mostrar en donde está esa caja (camión 1, camión 2, lote, o no está en transporte)

<u>Por ejemplo</u>, si se tienen estos datos,

Camión1	201 202 203 204 205 206
Camión 2	301 302 303 304 305
Lote_a_cargar	91 92 93 94 95 96 97

se mostrará por pantalla:

Ingrese opción (1) disponible (2) hay espacio (3) donde caja > 1
Pueden cargarse. Camión 1: 94 cajas, y camión 2: 95 cajas
Ingrese opción (1) disponible (2) hay espacio (3) donde caja > 2
El lote puede cargarse completamente
Ingrese opción (1) disponible (2) hay espacio (3) donde caja > 3
Ingrese código de caja a buscar: 94
La caja buscada está en el lote a cargar

Ejercicio 26 (importante: se evaluará la correcta división del problema en subproblemas y la elección adecuada de procedimientos o funciones para implementarlos).

En una biblioteca cada libro tiene un código único que está representado por un número entero. El sistema de la biblioteca dispone de 3 archivos de números enteros (ordenados de menor a mayor): el archivo "disponibles" con los códigos de los libros que están en la biblioteca y pueden ser prestados; el archivo "prestados" que tiene los códigos de los libros que fueron prestados y no están en la biblioteca; y el archivo "reservados" con los libros que fueron prestados, pero que además fueron reservados por alguien para sacarlos cuando sean devueltos.

Además, existe otro archivo llamado "pedidos" que tiene los códigos de los libros que quieren ser retirados de la biblioteca. Observe que puede haber archivos vacíos.

Escriba un programa en Pascal que lea del archivo "pedidos" y para cada código N encontrado, escriba en un archivo de texto una línea con el texto que se indica a continuación para cada caso.

"El libro de código N está disponible para su préstamo"

"El libro de código N está prestado y lamentablemente ya fue reservado"

"El libro de código N está prestado pero fue reservado"

"El libro de código N no existe en la biblioteca"

En el caso que un código N figure como "prestado" pero no está "reservado" entonces el programa deberá agregar ese código N al archivo de reservados.

Debe implementar toda primitiva que utilice.