

**Resolución de Problemas y Algoritmos**

**Clase 14**  
**Pascal: pasaje de parámetros y división de problemas usando archivos. Archivos de texto.**




**Dr. Alejandro J. García**  
 http://cs.uns.edu.ar/~ajg



Departamento de Ciencias e Ingeniería de la Computación  
 Universidad Nacional del Sur  
 Bahía Blanca - Argentina

**Problema propuesto**

- Realizar un programa que elimine todas las apariciones de un elemento en un archivo de enteros llamado 'mis-numeros.datos' .

Solución y división del problema:

- Leo el elemento E a eliminar
- Escribo en un auxiliar los elementos distintos a E
- Copio del auxiliar al original (primitiva)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

```

program eliminar;
type Tcomponente = Integer; Arch_int: FILE OF Tcomponente;
var F1,F2 : Arch_int; Eli: Tcomponente;
...
las primitivas eliminarE y copiar están a continuación
...
begin
  assign(F1, 'mis-numeros.datos');
  assign(F2, 'mis-numeros.copia');
  writeln('Ingrese elemento a eliminar');
  readln(eli);
  eliminarE(F1,F2,eli);
  copiar(F2,F1);
end.
    
```

Observe el orden en los parámetros efectivos

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

```

type Tcomponente = Integer; Arch_int: FILE OF Tcomponente;
PROCEDURE eliminarE( VAR original, copia: Arch_int;
                    eli: Tcomponente);
{... escriba en "copia" todos los que son distintos a "eli" ...}
Var elemento: Tcomponente;
begin
  Reset(original); rewrite(copia);
  while not eof(original) do begin
    read(original,elemento);
    if elemento <> eli then write(copia, elemento);
  end;
  close(original); close(copia);
End;
    
```

Con los archivos es obligatorio usar parámetros por referencia

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

```

type Tcomponente = Integer; Arch_int: FILE OF Tcomponente;
PROCEDURE Copiar( VAR origen, destino: Arch_int);
{... hace una copia idéntica del archivo origen en destino...}
var elemento: Tcomponente;
begin
  Reset(origen); rewrite(destino);
  while not eof(origen) do begin
    read(origen,elemento); write(destino, elemento);
  end;
  close(origen); close(destino);
End;
    
```

Con los archivos es obligatorio usar parámetros por referencia

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

**Problema propuesto como tarea**

Realizar un programa que muestre el contenido de un archivo de enteros llamado 'mis-numeros.datos', luego solicite al usuario un elemento E, elimine todas las apariciones E, y vuelva a mostrar el contenido del archivo. Esta operación podría repetirse cuantas veces el usuario quiera.

Obs: este problema tiene algunas partes similares al anterior. Para la solución puede usar primitivas del problema anterior y construir nuevas.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

### Archivos de texto en Pascal (TEXT)

- En Pascal, existe un TIPO PREDEFINIDO TEXT para archivos de texto.

**Program ejemplo;**

```
VAR T1, T2, documento: TEXT;
```

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    7

### Operaciones sobre archivos de texto en Pascal

- Además de las operaciones vistas sobre archivos secuenciales se agregan:

**Procedimientos predefinidos:**

- readln(T):** desplaza el “puntero” de lectura en T hasta el carácter siguiente a un fin de línea (enter).
- writeln(T):** escribe un fin de línea (enter) en T.

**Función predefinida:**

- eoln(F)** (end of line): retorna TRUE si se llegó al final de una línea y FALSE en caso contrario.

**Observaciones:**

**readln(T,e)** es equivalente a **read(T,e); readln(T)**  
**writeln(T,e)** es equivalente a **write(T,e); writeln(T)**

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    10

**Problema:** escriba un programa para abrir un archivo de texto ya existente llamado “texto.txt”, y contar cuantas líneas tiene.

**Program líneas;**

```
VAR T: TEXT; cant:integer;
begin
  assign(T, 'texto.txt'); reset(T); cant:=0;
  while not eof(T) do
    begin
      readln(T); cant:=cant+1;
    end;
  writeln('Cantidad de líneas: ', cant);
  close(T);
end.
```

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    11

**Problema:** escriba un programa para abrir un archivo de texto ya existente llamado “texto.txt”, y generar otro que cuando encuentre un punto baje de línea.

**Program líneas;**

```
VAR T1,T2: TEXT; cant:integer; ch: char;
begin
  assign(T1, 'texto.txt'); reset(T1);
  assign(T2, 'otro.txt'); rewrite(T2);
  while not eof(T1) do
    begin
      read(T1,ch);
      write(T2,ch);
      if ch='.' then writeln(T2);
    end;
  close(T1); close(T2);
end.
```

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    12

### Observación importante

- Los identificadores (constantes, tipos, variables, procedimientos, o funciones) del entorno **global** son visibles en **todo el programa**.
- En esta materia se permite el uso de constantes globales, procedimientos o funciones globales, y tipos globales..

**Pero...**

VARIABLES GLOBALES y NO LOCALES

- En esta materia se **PROHIBE** el uso de **VARIABLES GLOBALES (y NO LOCALES)** en procedimientos y funciones.
- Siempre que tenga la necesidad de usarlas, tiene que usar un parámetro.

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    18

### Tipos de datos en Pascal

(hasta ahora...) **Tipo de Dato:** Conjunto de valores posibles que puede tomar una variable....

**¿Qué elementos de Pascal tienen un tipo asociado?**

- una variable,
- una expresión,
- un parámetro,
- una función.

**Tipo de Dato:** define el conjunto de valores posibles que puede tomar una variable, expresión, parámetro o función; y también define las operaciones que pueden usarse sobre esos valores.

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    19

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

**Procedimientos y parámetros en Pascal**  
 MultiplicarFracciones tiene 6 parámetros: 4 por valor (para recibir datos) y 2 por referencia (para devolver datos)

```
PROCEDURE MultiplicarFracciones
  ( Num1, Den1, Num2, Den2: INTEGER;
    VAR NumRes, DenRes: INTEGER);
BEGIN
  NumRes := Num1 * Num2;
  DenRes := Den1 * Den2;
END;
```

¿Puede un procedimiento tener sólo parámetros por v?

```
PROCEDURE BAJAR_LINEAS(cant:INTEGER);
var v:integer;
BEGIN
  FOR v:=1 TO cant DO writeln;
```

END;

**Procedimientos y parámetros en Pascal**  
 ¿ Puede un procedimiento tener sólo parámetros por referencia?

```
PROCEDURE PasarAmayuscula(VAR L:char);
begin
  L := chr(ord(L) - (ord('a') - ord('A')));
end;
```

¿ Puede un procedimiento no tener parámetros?

```
PROCEDURE PAUSA;
BEGIN
  writeln(' pulse ENTER para continuar'); readln;
```

END;

¿Puede un procedimi. tener un único dato de salida?

```
PROCEDURE EsVocal (letra :char; var ES: boolean);
BEGIN
CASE letra OF
  'A','E','I','O','U', 'a','e','i','o','u': Es:=true;
ELSE Es:=false;
END;
```

¿Qué diferencia hay con tener una función?

```
FUNCTION EsVocal (letra :char): boolean;
BEGIN
CASE letra OF
  'A','E','I','O','U', 'a','e','i','o','u': EsVocal:=true;
ELSE EsVocal:=false;
END;
```

**Funciones y parámetros en Pascal**  
 ¿Puede una función no tener parámetros?

```
FUNCTION leer_letra:CHAR;
var aux: char;
BEGIN
  REPEAT
    read(aux)
  UNTIL (aux>='A') and (aux<='Z')
    or (aux>='a') and (aux<='z')
  leer_letra:= aux
END;
```

**Funciones y parámetros en Pascal**  
 ¿ Puede una función tener parámetros por referencia?

```
FUNCTION leer_letra(VAR error:boolean) :CHAR;
var aux: char;
BEGIN
  read(aux);
  IF (aux>='A') and (aux<='Z')
    or (aux>='a') and (aux<='z')
  then error:=FALSE
  else error:=TRUE;
  leer_letra:= aux
END;
```

**Preguntas teóricas frecuentes**

- ¿Cuándo un identificador es visible (referenciable) en un bloque? ¿Cuándo está oculto?
- ¿Qué elementos forman el entorno de referencia de un bloque?
- ¿Qué diferencia hay entre un parámetro por valor y uno por referencia?
- ¿Porqué es importante distinguir entre declarar un identificador y usar (referenciar) un identificador?
- ¿Qué diferencias hay entre implementar la primitiva **factorial** como una función o como un procedimiento?
- ¿ Toda función puede ser transformada a un procedimiento? ¿Qué gana? ¿Qué pierde?

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.