

## Resolución de Problemas y Algoritmos

### Clase 19:

#### Resolución de problemas utilizando recursión



**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

### Repaso

Considere los siguientes conceptos:

- Parámetros por valor y referencia
- Variables locales
- Parámetros efectivos

¿Qué relación existe y qué diferencias hay entre ellos?

Estos conceptos deben estar claros ya que serán parte de la implementación de funciones o procedimientos recursivos.

Antes de trabajar sobre nuevos ejemplos recursivos haremos algunos ejemplos sobre esos conceptos.

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

```

program Reflexion; {objetivo: reflexionar sobre parámetros y variables locales. }
var aux,efectivo:integer; {quedó un poco compacto para que entre en una "slide"}
procedure Xvalor (parametro:integer);
begin writeln('...Entro a Xvalor con ', parametro);
      parametro:=200;
      writeln('...Salgo de Xvalor con ', parametro); end;
procedure Xreferencia (var parametro: integer);
begin writeln('...Entro a Xreferencia con ', parametro);
      parametro:=200;
      writeln('...Salgo de Xreferencia con ', parametro); end;
procedure varLocal;
var aux: integer;
begin writeln('...Entro a varLocal con ', aux);
      aux:=200;
      writeln('...Salgo de varLocal con ', aux); end;
begin
efectivo:= 1; aux:=10; writeln('En el programa', efectivo,' y ',aux);
varLocal;      writeln('Después de llamar a varLocal: ',efectivo,'y',aux);
Xvalor(efectivo);      writeln('Después de llamar a Xvalor: ',efectivo);
Xreferencia(efectivo); writeln('Después de llamar a Xreferencia: ',efectivo);
end.
    
```

Tarea: Primero haga una traza en papel (bien prolija) y luego ejecute en su computadora para comparar. (Puede agregar mas "writeln"s.)

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

### Salida por consola del programa "reflexión"

```

En el programa: 1 y 10
...Entro a varLocal con 4223489
...Salgo de varLocal con 200
Después de llamar a varLocal : 1 y 10
...Entro a Xvalor con 1
...Salgo de Xvalor con 200
Después de llamar a Xvalor : 1
...Entro a Xreferencia con 1
...Salgo de Xreferencia con 200
Después de llamar a Xreferencia: 200
    
```

Las variables locales no tienen valor inicial, observe el "valor" de la variable aux cuando aún está sin valor inicial.

Los valores de efectivo y aux no cambian.

El valor de efectivo no cambia

El valor de efectivo cambia en este caso.

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

```

program Reflexion4; {El objetivo de este programa es hacer una traza y reflexionar sobre el pasaje de parámetros por referencia. }
var v1,v2,v3,v4:integer; {quedó un poco compacto para que entre en una "slide"}
procedure P3 (var R, C, X:integer; N:integer);
var local: integer;
begin writeln('Entro a P3 con ', R:9, N:9);
      local:= 3; N:= local+N; R:=N; C:=0; X:=R;
      writeln('Salgo de P3 con ', local, R:9, C:9, X:9, N:9); end;
procedure P2 (var R, C, X:integer; N:integer);
var local: integer;
begin writeln('Entro a P2 con ', R:9, N:9);
      local:= 2; P3 (local,C,X,N+1); R:=local+N; C:=C+1;
      writeln('Salgo de P2 con ', local, R:9, C:9, X:9, N:9); end;
procedure P1 (var R, C, X:integer; N:integer);
var local: integer;
begin writeln('Entro a P1 con ', R:9, N:9);
      local:= 1; P2 (local,C,X,N+1); R:=local+N; C:=C+1;
      writeln('Salgo de P1 con ', local, R:9, C:9, X:9, N:9); end;
begin v1:=5; v4:=1; P1(v1,v2,v3,v4); write('finalizo con ', v1,v2,v3,v4); end.
    
```

Tarea: Primero haga una traza en papel (bien prolija) y luego ejecute en su computadora para comparar. (Puede agregar mas "writeln"s.)

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

### Salida por consola de "reflexión4"

```

Entro a P1 con      5      1
Entro a P2 con      1      2
Entro a P3 con      2      3
Salgo de P3 con     3      6      0      6      6
Salgo de P2 con     6      8      1      6      2
Salgo de P1 con     8      9      2      6      1
Finalizo con        9      2      6      1
    
```

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) 2015**

**Preguntas (¿teóricas?) para reflexionar**

Las siguientes preguntas son sobre el programa "reflexion4",

1. ¿Cuáles son parámetros por referencia en P1?
2. La variable v2 no tiene valor al ser usada en el parámetro efectivo de la llamada a P1, ¿es un error de programación?
3. La variable v1 si tiene valor ¿es un error? ¿es mejor?
4. ¿Qué ocurriría si en P3 no se hiciera C:=0?

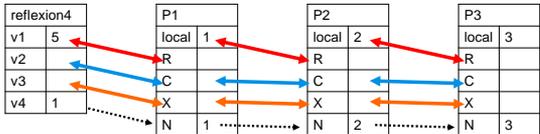
**Parte de la traza**

Estado de la traza antes de llamar a P1

reflexion4	
v1	5
v2	
v3	
v4	1

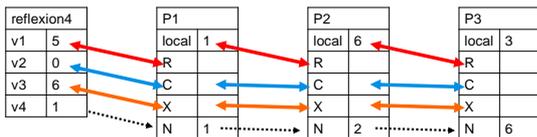
En esta página y las siguientes se muestran algunas partes de la traza del programa reflexion4. Sugerencia: realice su propia traza completa y compare.

Estado de la traza luego de ejecutar "local:= 3" en P3

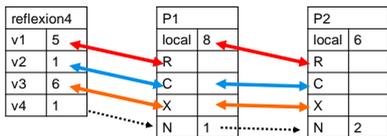


**Parte de la traza**

Luego de ejecutar "X:= R" en P3: observe los cambios en v2 y v3, (y local de P2)

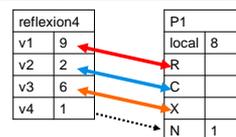


Luego de ejecutar "C:=C+1;" en P2: observe los cambios en v2 y v3



**Parte de la traza**

Luego de ejecutar "C:=C+1;" en P1: observe los cambios en v2 y v3



Estado final de las variables globales.

reflexion4	
v1	9
v2	2
v3	6
v4	1

**Problema propuesto: cantidad de elementos**

Escriba un planteo recursivo y luego un procedimiento que respete ese planteo para contar la cantidad de elementos de un archivo.

Ejemplos: 1 2 3 4 (4 elem.) 12 -34 (2 elem.)  
archivo vacío (0 elem.)

**Planteo:** cantidad de elementos de un archivo A

**Caso base:** si el archivo está vacío entonces la cantidad es 0 (cero)

**Caso general:** si el archivo no está vacío, entonces la cantidad es 1 + la cantidad de elementos del archivo A sin su primer elemento.

**Implementación en Pascal**

- Como fue dicho antes no hay una única forma de escribir un procedimiento que respete el planteo.
- Hay que tener cuidado donde realiza "assign", "reset" y "close" del archivo.
- **Pregunta teórica:** ¿necesita hacer una primitiva recursiva diferente para cada tipo de archivo? Escriba su respuesta y consulte sus dudas.
- **Tarea:** (para practicar) Realice una función recursiva que respete el planteo anterior y cuente la cantidad de elementos de un archivo.

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) 2015

```

program prueba1;
type Telemento = integer; Tarchi = file of Telemento;
var A: Tarchi; cantidad: integer;

Procedure contar (var F: Tarchi; var cant:integer);
{cuenta los elementos de un archivo}
var ele: telemento; aux:integer;
begin
  if EOF(F) then cant:=0 {caso base}
  else begin {caso general}
    read(F,ele); {leo el primero elemento}
    contar(F, aux); {llamo con F sin su primer elemento}
    cant:= aux +1;
  end;
begin
assign (A, 'el-archivo');
reset(A); contar(A, cantidad); close(A);
writeln('cantidad de elementos: ',cantidad);
end.
    
```

¿Qué pasaría si hago reset y close dentro del procedimiento recursivo?

### Observaciones

- En el programa anterior (prueba1) **assign**, **reset** y **close** del archivo se realizan en el bloque principal. Vea por ejemplo lo que pasa en estos dos casos que está mal implementado:

```

Procedure contar (var F: Tarchi; ...
{cuenta los elementos de un archivo}
var ele: telemento; aux:integer;
begin
reset(F);
if EOF(F) then ...
    
```

MAL

```

...
if EOF(F) then cant:=0
else begin {caso general}
  reset(F);
  read(F,ele);
  contar(F, aux);
  cant:= aux +1;
end;
    
```

MAL

- En cualquiera de los dos ejemplos anteriores cada vez que se llama recursivamente se ejecuta nuevamente **reset(F)**, con lo cual se vuelve a comenzar a leer del primer elemento y se produce una ejecución infinita, ya que nunca se reduce el archivo en un elemento (no respeta el planteo).

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

### Observaciones

- En el programa anterior (prueba1), en el procedimiento recursivo "contar" la variable local "aux" es utilizada para almacenar la cantidad de elementos del "archivo sin su primer elemento".
- Realice la traza y verá que en cada llamada recursiva "aux" recibe la cantidad calculada por la invocación recursiva y luego el parámetro por referencia "cant" retorna "aux" + 1 a quien lo llamó.
- En el programa siguiente (prueba2) hay otra versión correcta del procedimiento recursivo que también respeta el planteo pero no usa "aux". Realice una traza para ver la diferencia en ejecución.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 15

```

program prueba2;
type Telemento = integer; Tarchi = file of Telemento;
var A: Tarchi; cantidad: integer;

Procedure contar (var F: Tarchi; var cant:integer);
var ele: telemento; {cuenta los elementos de un archivo}
begin
  if EOF(F) then cant:=0 {caso base}
  else begin read(F,ele); {caso recursivo}
    contar(F,cant);
    cant:=cant+1;
  end;
begin
assign (A, 'el-archivo');
reset(A); contar(A, cantidad); close(A);
writeln('cantidad de elementos: ',cantidad);
end.
    
```

Observe que cambia en la traza si no usa la variable local "aux"

### Observaciones

- En el programa siguiente (prueba3) hay otra versión correcta del procedimiento recursivo que también respeta el planteo.
- En este caso contar abre y cierra el archivo.
- Para hacer esto tiene su propio procedimiento interno que hace la tarea recursiva.
- Realice una traza para ver la diferencia en ejecución.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 17

```

program prueba3;
type Telemento = integer; Tarchi = file of Telemento;
var A: Tarchi; cantidad: integer;

Procedure contar (var F: Tarchi; var cant:integer);
Procedure contar_rec (var F: Tarchi; var cant:integer);
var ele: telemento; {cuenta los elementos de un archivo}
begin
  if eof(F) then cant:=0 {caso base}
  else begin read(F,ele); {caso recursivo}
    contar_rec(F,cant);
    cant:=cant+1;
  end;
begin
  {abre el archivo, llama al recursivo y cierra el archivo}
  reset(F); contar_rec(F, cantidad); close(F);
end;
Begin assign (A, 'el-archivo'); contar(A, cantidad);
writeln('cantidad de elementos: ',cantidad); end.
    
```

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

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) 2015

Continuará

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) 2015**