

## Resolución de Problemas y Algoritmos

### Clase 17

#### 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

### Llamadas entre procedimientos

```

PROGRAM LLAMADAS;

PROCEDURE UNO (I: INTEGER);
BEGIN
    writeln(' Estoy en uno ',I);
END;

PROCEDURE DOS (K: INTEGER);
BEGIN
    writeln(' Estoy en dos ' , K);
END;

BEGIN
    UNO(1);
    DOS(2);
END.
    
```

¿Puedo llamar a DOS desde acá?  
¿Puedo llamar a UNO desde acá?  
¿Qué identificadores son visibles en cada uno de los procedimientos?

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

### Dos llama a uno

```

PROGRAM LLAMADAS;

PROCEDURE UNO (I: INTEGER);
BEGIN
    writeln(' Estoy en uno ',I);
END;

PROCEDURE DOS (K: INTEGER);
BEGIN
    writeln(' Estoy en dos ' , K);
    UNO(10);
END;

BEGIN
    UNO(1);
    DOS(2);
END.
    
```

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

### Uno llama a Dos: uso de "forward"

```

PROGRAM LLAMADAS;

PROCEDURE DOS (K: INTEGER); FORWARD;

PROCEDURE UNO (I: INTEGER);
BEGIN
    writeln(' Estoy en uno ',I);
    DOS(10);
END;

PROCEDURE DOS (K: INTEGER);
BEGIN
    writeln(' Estoy en dos ' , K);
END;

BEGIN
    UNO(1);
    DOS(2);
END.
    
```

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

### Ejemplo de un error escondido

```

PROGRAM ojo; {Este programa tiene un error en ejecución}
var r:real;
FUNCTION exp(base,exp:integer):integer;
VAR i,resultado:integer;
BEGIN
    resultado:=1;
    FOR i:=1 to exp DO resultado:=base*resultado;
    writeln('en la función el resultado es: ',resultado); readln;
    exp:= resultado;
END;
BEGIN
    r:=exp(2,3);
    writeln('Resultado de 2 a la 3: ', r:0:0);
END.
    
```

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

### Repaso de teoría: Entornos de referencia

- Recuerde que un identificador de nombre N en un entorno local **oculta** a todo identificador del mismo nombre N en otro entorno (no-local, global, predefinido).
- Por ejemplo, en el programa anterior, el nombre de la función "exp" pertenece al entorno global.
- El mismo identificador "exp" fue utilizado (erróneamente) como parámetro en la función. Entonces, el parámetro "exp" que pertenece al entorno local de la función, oculta al "exp" global.
- Con lo cual la asignación "exp:=resultado", está modificando el parámetro en lugar de retornar el resultado de la función.

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.

### Problema propuesto

- Escriba un programa que permita ingresar por teclado una secuencia de caracteres terminada en un punto (por ejemplo: “hola que tal.”) y que la muestre por pantalla en orden inverso (“lat euq aloh”).
- Ejemplo:

Ingrese una cadena terminada en punto: un animal.  
Invertida queda así: lamina nu

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

```

program Invertir_Recursivo;
  Procedure InvertirEntrada;
  var caract: char;
  begin
    read(caract);
    IF caract = '.' THEN write(' Invertida queda así: ')
    ELSE begin
      InvertirEntrada;
      write(caract);
    end;
  end;
begin
  writeln(' Ingrese una cadena terminada en punto: ');
  InvertirEntrada;
end.
    
```

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    8

### Planteo Recursivo para buscar elemento

- Escribir una función recursiva para buscar un elemento en un archivo secuencial.

**Ejemplos:** 3 está en 1 2 3 y en 3 2 1 3 2, pero 3 no está en 1 2 4 y en el archivo vacío.

**Buscar el elemento X en la secuencia S:**

- Caso base: Si la secuencia es vacía retornar falso.
- Caso general : Si la secuencia tiene más de un elemento entonces: si el primer elemento de S es X retornar verdadero, de lo contrario buscar el elemento X en la secuencia S' que resulta de S sin su primer elemento.

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    9

### Función Recursiva para buscar elemento

```

Type archiEnt = file of integer;
Function esta(buscado:integer; var F: archiEnt):boolean;
var elemento: integer;
begin
  IF EOF(F) THEN esta:=false
  ELSE begin
    read(F,elemento);
    IF buscado = elemento then esta:= true
    else esta:=esta(buscado,F);
  end;
end;
    
```

- A continuación se muestran tres alternativas de cómo incluir en el programa esta función.
- Observe que el “reset” del archivo NO puede estar dentro de la función recursiva.

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

```

program Buscar1; {busca un elemento en un archivo}
Type archiEnt = file of integer;
var elem: integer; F: archiEnt;
Function esta(buscado:integer; var F: archiEnt):boolean;
begin
  Assign(F,'miarchivo.int');
  Reset(F);
  Writeln('ingrese elemento a buscar'); readln(elem);
  If esta(elem,F) then writeln(' fue encontrado')
  else writeln('no fue encontrado');
  close(F);
end.
    
```

Esta alternativa es útil cuando simplemente se quiere probar la función y no es parte de un problema más complejo. No obstante, se puede hacer mejor.

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

```

program Buscar2; {una función específica llama a esta()}
Type archiEnt = file of integer; var x: integer; F: archiEnt;
Function esta(buscado:integer; var F: archiEnt):boolean;
Function estaXenF(X: integer; var F: archiEnt):boolean;
Begin
  Reset(F); estaXenF:=esta(X,F); close(F);
end
Begin Assign(F,'miarchivo.int');
Writeln('ingrese elemento a buscar'); readln(elem);
If estaXenF(elem,F) then writeln(' fue encontrado')
else writeln('no fue encontrado');
end.
    
```

- Esta versión es mejor que la anterior si la primitiva estaXenF() es usada como parte de la división de un problema más complejo. Pero... (sigue)

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

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.

### Comentarios sobre buscar2

- En la alternativa planteada en buscar2, tanto esta() como estaXenF() están en el entorno global y (de no ser ocultadas por otro identificador más local) pueden ser llamadas desde cualquier otra parte del programa.
- Esto podría traer problemas si el programador (o alguien del grupo de trabajo) invoca a la primitiva esta() directamente, ya que el archivo no será abierto para lectura, y podría generar un error en ejecución.
- Para evitar esto se puede ubicar a esta() “dentro” de la función estaXenF().

```

program Buscar3; {la función recursiva está encapsulada}
Type archiEnt = file of integer; var x: integer; F: archiEnt;

Function estaXenF(X:integer; var F: archiEnt):boolean;
Function esta(X:integer; var F: archiEnt):boolean;
Begin
Reset(F); estaXenF:=esta(X,F); close(F);
end
Begin
...
end.
    
```

- Esta alternativa tiene al ventaja adicional que la función esta() es parte del entorno local de estaXenF, y por lo tanto, no se podrá llamar directamente a esta() desde otra parte del programa que no sea la función estaXenF.

### Problema propuesto

- Escriba un programa con una función recursiva que indique cuantas veces está un elemento en un archivo.

Cantidad de veces que está X en secuencia S :

- Caso base: Si la secuencia es vacía retornar 0.
- Caso general : Si la secuencia tiene más de un elemento entonces: Sea C la cantidad veces que esta X en la secuencia S' que resulta de S sin su primer elemento. Si el primer elemento de S es X entonces la cantidad de veces es C + 1, de lo contrario la cantidad es C.

### Función Recursiva para contar elemento

```

Type archiEnt = file of integer;

Function cuantas(X:integer; var F: archiEnt):integer;
var aux, c: integer;
begin
IF EOF(F) THEN cuantas:=0
ELSE begin
read(F,aux);
c:=cuantas(X,F);
IF X = aux then cuantas:= c +1
else cuantas:= c;
end;
end;
end;
    
```

- Elija una alternativa de las 3 mencionadas antes, escriba un programa que llame a la función, y realice una traza.

### Problema propuesto

Escriba un programa con una función recursiva que indique cuantas veces está un elemento en un archivo.

#### Generar la secuencia invertida de S

- Caso base: Si S es vacía, la secuencia invertida es vacía.
- Caso general : si la secuencia tiene elementos, retirar el primer elemento (P); generar la secuencia invertida de S sin su primer elemento. Agregar P al final.

A continuación se muestra una implementación con procedimientos anidados.

```

Procedure invertir( var F1,F2: archiEnt);
Procedure invertirRec( var F1,F2: archiEnt);
var aux: integer;
begin
IF not EOF(F) THEN
begin
read(F,aux);
invertirRec(F1,F2);
write(F2,aux);
end;
end;
Begin
Reset(F1); rewrite(F2);
invertirRec(F1,F2);
close(F1); close (F2);
end
    
```

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.