

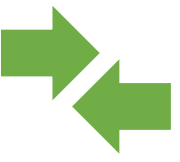


# Conceptos de Algoritmos Datos y Programas

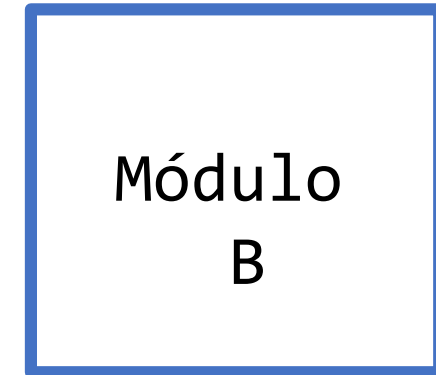
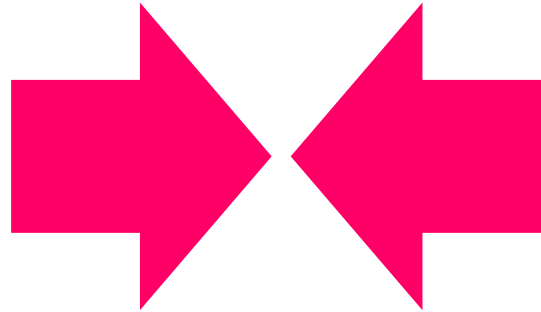
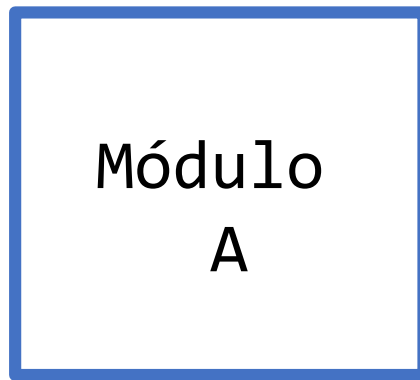
# CADP – Temas de la clase de hoy



- Comunicación entre módulos

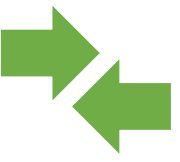


## COMUNICACIÓN ENTRE MODULOS



**Variables Globales**  
**Parámetros**

*Cuál  
utilizamos?*



## VARIABLES GLOBALES

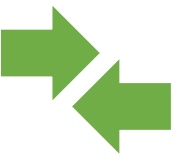
Demasiados identificadores

No se especifica la comunicación entre los módulos

Conflictos de nombres de identificadores utilizados por diferentes programadores.

Posibilidad de perder integridad de los datos, al modificar involuntariamente en un módulo datos de alguna variable que luego deberá utilizar otro módulo.



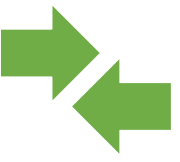


## PARAMETROS

La solución a estos problemas ocasionados por el uso de variables globales es una combinación de **ocultamiento de datos (Data Hiding )** y **uso de parámetros**.

El ocultamiento de datos significa que los datos exclusivos de un módulo NO deben ser "visibles" o utilizables por los demás módulos.

El uso de parámetros significa que los datos compartidos se deben especificar como parámetros que se transmiten entre módulos.



## PARAMETROS – Cómo vamos a trabajar?

1

- Se analiza para cada módulo entonces: ¿cuáles son los datos propios? y ¿cuáles son los datos compartidos?

2

- Los datos propios se declararan locales al módulo.

3

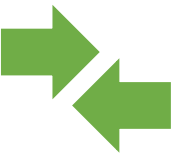
- Los datos compartidos se declararán como parámetros.



Parámetros por valor



Parámetros por referencia



## PARAMETRO POR VALOR

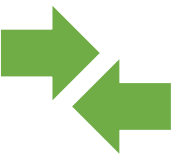
Un dato de entrada por valor es llamado parámetro IN y significa que el módulo recibe (sobre una variable local) un valor proveniente de otro módulo (o del programa principal).

Con él puede realizar operaciones y/o cálculos, pero no producirá ningún cambio ni tampoco tendrá incidencia fuera del módulo.

*Con qué tipo de parámetro se relaciona?*

*Cómo se declaran?*

*Cómo se usan?*



## PARAMETRO POR VALOR

```
procedure uno (nombre1: tipo; nombre2: tipo);
```

```
var
```

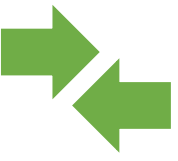
```
...
```

```
Begin
```

```
    Uso de los parámetros con nombre1 y nombre2
```

```
End;
```





## PARAMETRO POR VALOR

Program porValor;

```
procedure uno (num: integer);
```

```
Begin
```

```
    if (num = 7) then
```

```
        num := num + 1;
```

```
    write (num);
```

```
end;
```

```
var
```

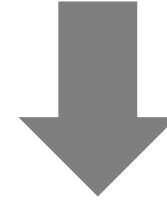
```
    x: integer;
```

```
begin
```

```
    x := 7;
```

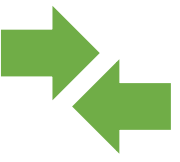
```
    uno (x);
```

```
end.
```



Dentro del procedimiento uno, el parámetro **num** copia el valor enviado por **x** (variable del programa)

**Cómo funciona?**



Program porValor;

```
procedure uno (num: integer);
```

```
Begin
```

```
    if (num = 7) then
```

```
        num:= num + 1;
```

```
    write (num);
```

```
end;
```

```
var
```

```
    x: integer;
```

```
begin
```

```
    x:= 7;
```

```
    uno (x);
```

```
end.
```



Qué pasa si después de  
llamar al procedimiento uno  
en el programa imprimo  
num?

Procedimiento uno  
Variables locales  
Parámetros

Programa ppal  
Variables globales  
Variables de prog

num = 8

Imprime 8

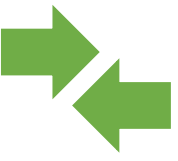
x = 7

Imprime 7

**MEMORIA**

# CADP – MODULARIZACION

## COMUNICACION



Program porValor;

```
procedure uno (num: integer);
```

```
Begin
```

```
    if (num = 7) then
```

```
        num:= num + 1;
```

```
    write (num);
```

```
end;
```

```
var
```

```
    num: integer;
```

```
begin
```

```
    num:= 7;
```

```
    uno (num);
```

```
end.
```



Qué pasa si después de  
llamar al procedimiento uno  
en el programa imprimo  
num?

Procedimiento uno  
Variables locales  
Parámetros

Programa ppal  
Variables globales  
Variables de prog

num = 8

Imprime 8

num = 7

Imprime 7

**MEMORIA**

# CADP – MODULARIZACION

Program porValor;

**procedure** uno (x: integer);

Begin

**if** (x = 7) **then**

        x := x + 1;

**write** (x);

**end**;

**var**

    x: integer;

**begin**

    x := 7;

    uno (x);

**end**.



**Qué valores  
imprimen?**

# COMUNICACION

Program porValor;

**procedure** uno (num: integer);

Var

    x: integer;

Begin

**if** (num = 7) **then**

        num := num + 1;

    x := num;

**write** (num); **write** (x);

**end**;

**var**

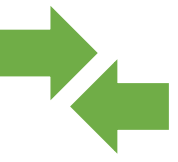
    x: integer;

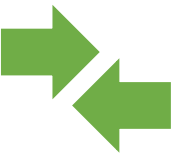
**begin**

    x := 7;

    uno (x);

**end**.





## PARAMETRO POR REFERENCIA

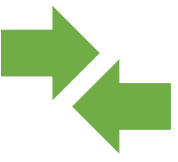
La comunicación por referencia (OUT, INOUT) significa que el módulo recibe el nombre de una variable (referencia a una dirección) conocida en otros módulos del sistema.

Puede operar con ella y su valor original dentro del módulo, y las modificaciones que se produzcan se reflejan en los demás módulos que conocen la variable.

*Con qué tipo de  
parámetro se  
relaciona?*

*Cómo se  
declaran?*

*Cómo se  
usan?*



## PARAMETRO POR REFERENCIA

```
procedure uno (var nombre1: tipo; var nombre2: tipo);
```

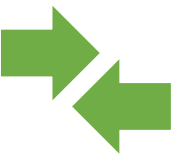
```
    var
```

```
        ...
```

```
    Begin
```

```
        Uso de los parámetros con nombre1 y nombre2
```

```
    End;
```



## PARAMETRO POR REFERENCIA

Program porReferencia;

```
procedure uno (var num: integer);
```

```
Begin
```

```
    if (num = ...) then
```

```
        num:= num + 1;
```

```
    write (num);
```

```
end;
```

```
var
```

```
    x: integer;
```

```
begin
```

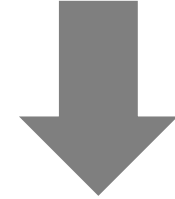
```
    x:= 7;
```

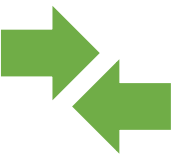
```
    uno (x);
```

```
end.
```

Dentro del procedimiento uno, el parámetro **num** comparte la dirección de memoria con **x** (variable del programa)

**Cómo funciona?**





Program porValor;

**procedure** uno (var **num**: integer);

Begin

    if (num = 7) then

        num := num + 1;

        write (num);

end;

var

**x**: integer;

begin

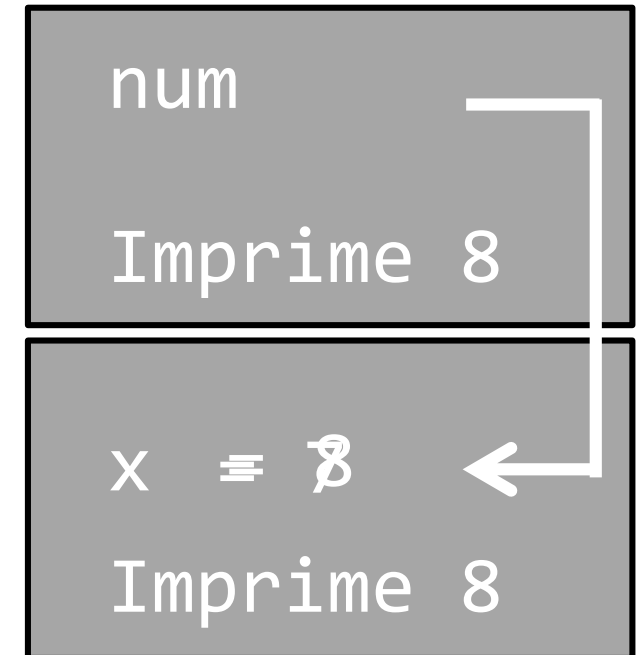
    x := 7;

    uno (x);

end.

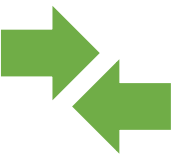
Procedimiento uno  
Variables locales  
Parámetros

Programa ppal  
Variables globales  
Variables de prog



**MEMORIA**





Program porValor;

**procedure** uno (var **num**: integer);

Begin

    if (num = 7) then

        num := num + 1;

        write (num);

end;

var

**num**: integer;

begin

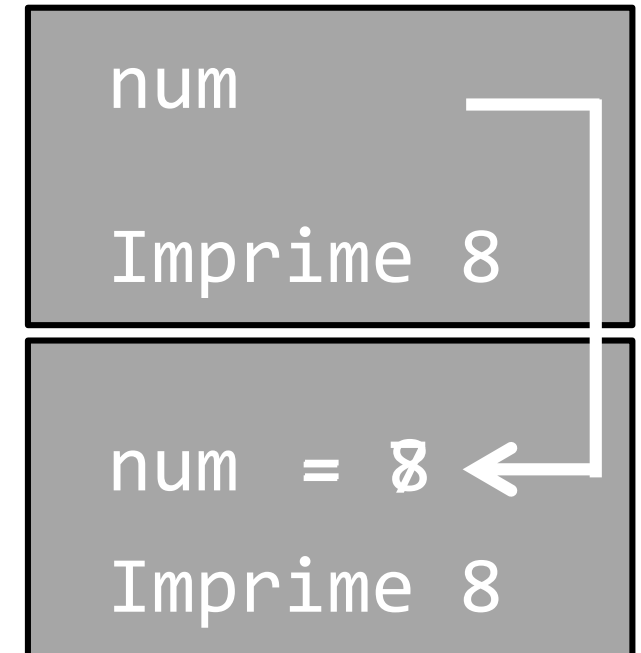
    num := 7;

    uno (num);

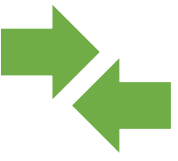
end.

Procedimiento uno  
Variables locales  
Parámetros

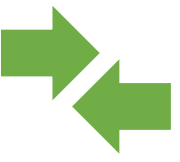
Programa ppal  
Variables globales  
Variables de prog



**MEMORIA**



- El número y tipo de los argumentos utilizados en la invocación a un módulo deben coincidir con el número y tipo de parámetros del encabezamiento del módulo.
- Un parámetro por valor debiera ser tratado como una variable de la cuál el módulo hace una copia y la utiliza localmente. Algunos lenguajes permiten la modificación local de un parámetro por valor, pero toda modificación realizada queda en el módulo en el cual el parámetro es utilizado.
- El número y tipo de los argumentos utilizados en la invocación a un módulo deben coincidir con el número y tipo de parámetros del encabezamiento del módulo.



Es  
correcto?

```
Program uno;
```

```
Var
```

```
    x:integer;
```

```
    c:char;
```

```
procedure ejemplo (var a:integer; j:char);  
begin
```

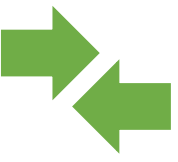
```
    // código del procedimiento ejemplo  
end;
```

```
begin
```

```
    c:='a';
```

```
    ejemplo (15,c);
```

```
end.
```



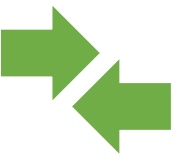
Es  
correcto?

```
Program uno;
```

```
Var  
    x:integer;
```

```
procedure ejemplo (var a:integer; j:char);  
begin  
    // código del procedimiento ejemplo  
end;
```

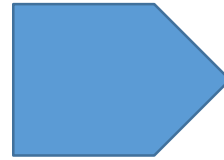
```
begin  
    x:=25;  
    ejemplo (x,'p');  
end.
```



Escriba un programa que lea un número entero e imprima la suma de todos sus dígitos. Realice dos soluciones:

- a) Modularice utilizando un procedimiento
- b) Modularice utilizando una función

4367

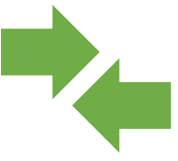


Imprime 20

- Qué es lo que se modulariza?
- Qué parámetros necesita el módulo?
- Cómo descompongo el número?

# CADP – MODULARIZACION

# COMUNICACION



```
procedure descomponer (num:integer; var suma:integer);
```

```
    num = 4367
```

```
    suma = 0
```

num = 4367

Tomo el último dígito = 7, lo sumo a suma, suma= 7

Con qué operación  
obtengo el último dígito?

Me deshago del último dígito  
Con qué operación  
deshecho el último dígito?

num = 463

Tomo el último dígito = 3, lo sumo a suma, suma= 10

Me deshago del último dígito

num = 46

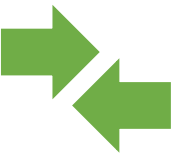
Tomo el último dígito = 6, lo sumo a suma, suma= 16

Me deshago del último dígito

num = 4

Tomo el último dígito = 4, lo sumo a suma, suma= 20

Me deshago del último dígito  
Con qué valor queda el num?



```
program uno;  
procedure descomponer (num:integer; var suma:integer);  
var
```

```
...
```

```
begin
```

```
...
```

```
end;
```

```
Var
```

```
valor,sum:integer;
```

```
Begin
```

```
read (valor);
```

```
descomponer (valor,sum);
```

```
write (sum);
```

```
End.
```

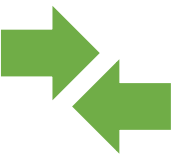


Copia el valor  
de valor



Comparte la  
dirección con sum

*Cómo se  
implementa  
descomponer?*



```
procedure descomponer (num:integer; var suma:integer);  
var  
  dig:integer;
```

Copia el  
valor recibido

Devuelve la suma

```
Begin
```

```
  suma:=0;
```

```
  while (num <> 0) do
```

```
    begin
```

```
      dig:= num MOD 10;
```

```
      suma:= suma + dig;
```

```
      num:= num DIV 10;
```

```
    end;
```

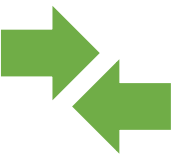
```
end;
```

Obtengo el último dígito

Descarto el último dígito

*Y con una  
función?*





```
function descomponer (num:integer):integer;
```

```
var
```

```
  dig,suma:integer;
```

```
Begin
```

```
  suma:=0;
```

```
  while (num <> 0) do
```

```
    begin
```

```
      dig:= num MOD 10;
```

```
      suma:= suma + dig;
```

```
      num:= num DIV 10;
```

```
    end;
```

```
    descomponer:= suma;
```

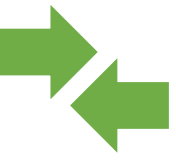
```
end;
```

→ Tipo de datos  
que devuelve

→ Copia el  
valor recibido

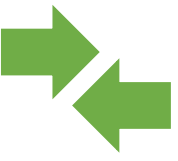
Por qué uso  
una variable  
suma?

Cómo escribo  
el programa?



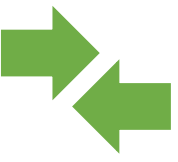
```
program uno;  
function descomponer (num:integer):integer;  
var  
    ...  
begin  
    ...  
end;  
  
Var  
    valor,sum:integer;  
Begin  
    read (valor);  
    sum:= descomponer (valor);  
    write (sum);  
End.
```

*Otra forma de  
invocar a la  
función  
descomponer?*

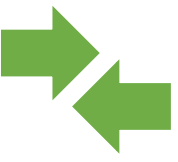


```
program uno;  
function descomponer (num:integer):integer;  
var  
    ...  
begin  
    ...  
end;  
  
Var  
    valor:integer;  
Begin  
    read (valor);  
    write (descomponer (valor));  
End.
```

*Qué modifico si se  
leen números  
hasta leer el  
número 50?*



```
program uno;  
function descomponer (num:integer):integer;  
begin  
    ...  
end;  
  
Var  
    valor:integer;  
Begin  
    read (valor);  
    while (num <> 50) do  
        begin  
            write (descomponer (valor));  
            read(valor);  
        end;  
    End.  
End.
```

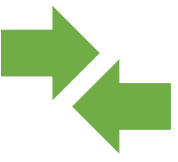


Escriba un programa que lea números enteros hasta leer el valor 50. Para cada número par leído informar el mayor y el menor dígito

4367	➡	No imprime nada por ser impar
24	➡	Imprime 2 imprime 4
1382	➡	Imprime 1 imprime 8
50	➡	Termina

● Qué es lo que se modulariza?

● Con qué tipo de módulos?



Escriba un programa que lea números enteros hasta leer el valor 50. Para cada número par leído informar el mayor y el menor dígito

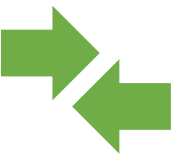
4367 → No imprime nada por ser impar

24 → Imprime 2 imprime 4

1382 → Imprime 1 imprime 8

50 → Termina

```
Leo un número (num)
While (num <> 50) do
begin
    if (el nro leído es par) then
        calculo menor y mayor dígito
        informa el menor y mayor dígito
    leo un número (num)
end.
```



```
function esPar (n:integer): boolean;  
Var  
    ok:boolean;  
Begin  
    if (n MOD 2 = 0) then ok:= true  
    else ok:= false;  
    esPar:= ok;  
end;
```

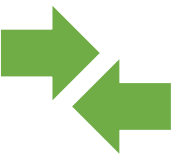
*Opción 1*

*Opción 2*

```
function esPar (n:integer): boolean;  
Var  
    ok:Boolean;  
Begin  
    ok:= (n MOD 2 = 0);  
    esPar:= ok;  
end;
```

*Opción 3*

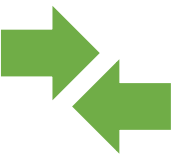
```
function esPar(n:integer):boolean;  
Begin  
    esPar:= (n MOD 2 = 0);  
end;
```



```
program uno;  
function esPar (n:integer):boolean;  
begin  
    esPar:= (n MOD 2 = 0);  
end;
```

```
var  
    num:integer;  par:boolean;  
begin  
    read (num);  
    while (num <> 50) do  
        begin  
            par:= esPar(num);  
            if (par = true) then  
                calculo digito máximo y mínimo  
                imprimo el dígito máximo y el dígito mínimo  
            read(num);  
        end;  
    end.
```

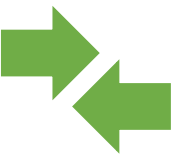




```
program uno;  
function esPar (n:integer):boolean;  
begin  
    esPar:= (n MOD 2 = 0);  
end;
```

```
var  
    num:integer;  
begin  
    read (num);  
    while (num <> 50) do  
        begin  
            if (esPar(num) = true) then  
                calculo digito máximo y mínimo  
                imprimo el dígito máximo y el dígito mínimo  
            read(num);  
        end;  
    end.
```

*Para el dígito  
máximo y mínimo  
que elijo una  
función o un  
procedimiento?*



```
Procedure maxmin( num:integer; var max,min:integer);
```

```
Var
```

```
    resto:integer;
```

```
Begin
```

```
    max:=0; min:=9;
```

```
    while (num <> 0) do
```

```
        begin
```

```
            resto:= num MOD 10;
```

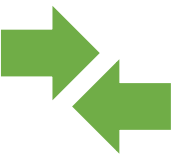
```
            if (resto >= max) then max:= resto;
```

```
            if (resto <= min) then min:= resto;
```

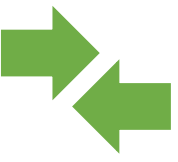
```
            num := num DIV 10;
```

```
        end;
```

```
End;
```

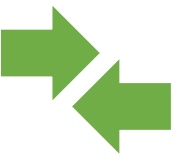


```
function maximo(num:integer):integer;  
Var  
    resto:integer;  
  
Begin  
    max:=0;  
    while (num <> 0) do  
        begin  
            resto:= num MOD 10;  
            if (resto >= max) then max:= resto;  
            num := num DIV 10;  
        end;  
    máximo:= max;  
End;
```



```
function minimo(num:integer):integer;  
Var  
    resto:integer;  
  
Begin  
    min:=9;  
    while (num <> 0) do  
        begin  
            resto:= num MOD 10;  
            if (resto <= min) then min:= resto;  
            num := num DIV 10;  
        end;  
    minimo:= min;  
End;
```

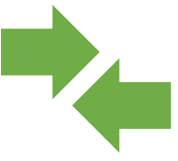
*Cómo escribimos  
el programa?*



## Cuál elijo?

```
Var
  num,maxi,mini:integer;
Begin
  read (num);
  while (num <> 50) do
    begin
      if (esPar (num)) then begin
        maxmin (num, maxi,mini);
        Write ('Dig máx', maxi);
        Write ('Dig mín', mini);
      end;
      read (num);
    end;
  end.
```

```
Var
  num,maxi,mini:integer;
Begin
  read (num);
  while (num <> 50) do
    begin
      if (impar (num) ) then begin
        maxi:= maximo(num);
        mini:= minimo(num);
        Write ('Dig máx', maxi);
        Write ('Dig mín', mini);
      end;
      read (num);
    end;
  end.
```



Program uno;

```
function esPar (n:integer):boolean;  
begin  
    esPar:= (n MOD 2 = 0);  
end;
```

```
Procedure maxmin( num:integer; var max,min:integer);
```

Var

```
    resto:integer;
```

Begin

```
    max:=0; min:=9;  
    while (num <> 0) do  
    begin  
        resto:= num MOD 10;  
        if (resto >= max) then max:= resto;  
        if (resto <= min) then min:= resto;  
        num := num DIV 10;  
    end;
```

End;

Var

```
    num,max,min:integer;
```

Begin

```
    read (num);
```

```
    while (num <> 50) do
```

```
        begin
```

```
            if (esPar (num)) then begin
```

```
                maxmin (num, maxi,mini);
```

```
                Write ('Dig máx', maxi);
```

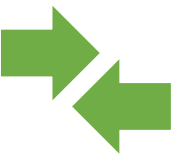
```
                Write ('Dig mín', mini);
```

```
            end;
```

```
            read (num);
```

```
        end;
```

```
    end.
```



```
Program uno;
```

```
Procedure numero (a: integer; var b:integer);
```

```
Var
```

```
  a:integer;
```

```
Begin
```

```
  a:= (36 MOD 2) + 3;
```

```
  if (a >= 3) then b:= b + a * 2  
                    else b:= b + a * 3;
```

```
  b:= (b * 6) DIV 5;
```

```
  write (a); write (b);
```

```
End;
```

```
Var
```

```
  a,b:integer;
```

```
Begin
```

```
  a:= 8;  b:= 6;
```

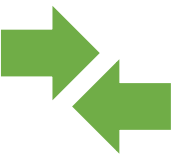
```
  numero (b,a);
```

```
  write (a); write (b);
```

```
End.
```



Qué  
imprime?

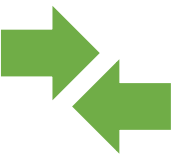


```
Program uno;  
Procedure numero (var a: integer; b:integer; var c:integer);  
Begin  
  a:= (b + c) MOD 2;  
  if (a MOD 2 = 0) then  
    begin  
      b:= b + a * 3;  
      c:= 8 DIV 2;  
      a:= a + b * c;  
    end  
  else begin  
    b:= (b + a) DIV 2;  
    c:= 25 DIV 4;  
    a:= a + b * c;  
  end;  
  write (a); write (b); write (c);  
End;  
  
Var  
  a,b,c:integer;  
Begin  
  a:= 18;  b:= 16;  c:= 3;  
  numero (c,a,b);  
  write (a); write (b); write (c);  
End.
```



Qué  
imprime?





```
Program uno;
Procedure numero (var a: integer; b:integer; var c:integer);
Var
  a:integer;
Begin
  a:= (b + c) MOD 2;
  if (a MOD 2 = 0) then
    begin
      b:= b + a * 3;    c:= 8 DIV 2;    a:= a + b * c;
    end
  else begin
    b:= (b + a) DIV 2;  c:= 25 DIV 4;  a:= a + b * c;
  end;
  write (a); write (b); write (c);
End;

Var
  a,b,c:integer;
Begin
  a:= 18;  b:= 16;  c:= 3;
  numero (c,a,b);
  write (a); write (b); write (c);
End.
```



Qué  
imprime?