

# **Introducción a la Programación**

## **Grado en Ingeniería Informática**

**Seminarios - Curso 2015-2016**

---

**Seminario 5**

## Ejercicio 1.- Enunciado

---

Diseñe una función factorial que reciba un número mayor o igual que cero y devuelva el factorial del número

## Ejercicio 1.- Pseudocódigo

Algoritmo Ejercicio factorial

Principal

var

entero: fat\_num, num

inicio

repetir

escribir("Introd. Num $\geq$ 0)

leer (num)

hasta\_que (num $\geq$ 0

fat\_num  $\leftarrow$  factorial(num)

escribir ("Fact =", fat\_num)

finprincipal

//cabecera: entero factorial (E ntero:n)

//precondicion n $\geq$ 0

//postcondicion devuelve factorial

entero factorial (E entero:n)

var

entero :fact

inicio

fact  $\leftarrow$  1

mientras ( n > 1) hacer

fact  $\leftarrow$  fact\*n

n  $\leftarrow$  n-1

finmientras

devolver fact

finfuncion

fin algoritmo

## Ejercicio 1.- Pseudocódigo

Algoritmo Ejercicio factorial

Principal

var

entero: fat\_num, num

inicio

repetir

escribir("Introd. Num $\geq$ 0)

leer (num)

hasta\_que (num $\geq$ 0

fat\_num  $\leftarrow$  factorial(num)

escribir ("Fact =", fat\_num)

finprincipal

//cabecera: entero factorial (E ntero:n)

//precondicion n $\geq$ 0

//postcondicion devuelve factorial

entero factorial (E entero:n)

var

entero :fact

inicio

fact  $\leftarrow$  1

mientras ( n > 1) hacer

fact  $\leftarrow$  fact\*n

n  $\leftarrow$  n-1

finmientras

devolver fact

finfuncion

fin algoritmo

## Ejercicio 1.-Lenguaje C

```
#include <stdio.h>
unsigned factorial (int n);
int main(){
    int num;
    unsigned resp;
    do{
        puts ("Introd. un n° positivo");
        scanf("%d",&num);
    }while (num < 0);
    resp=factorial (num);
    printf("El fact de %d es
           %u\n",num,resp);
    return 0;
}
```

/\*cabecera: unsigned factorial (int  
n)  
precondicion n>=0  
postcondicion devuelve factorial\*/  
unsigned factorial (int n){  
 unsigned fact=1;  
 while (n > 1)  
 fact \*=n--;  
 return fact;  
}

Diseña una función para la siguiente especificación

```
//cabecera void factorial ( int n, unsigned *fact)  
//precondición  $n \geq 0$   
//postcondición retorna en fact el calculo de factorial de n
```

## Ejercicio 2.-Lenguaje C

```
#include <stdio.h>

void factorial (int n, unsigned *);

int main(){
    int num;
    unsigned resp;
    do{
        puts ("Introd. un n° positivo");
        scanf("%d",&num);
    }while (num < 0);
    factorial (num, &resp);
    printf("El fact de %d es
           %u\n",num,resp);
    return 0;
}
```

**/cabecera** void factorial ( int n,  
unsigned \*fact)  
**//precondición** n >=0  
**//postcondición** retorna en fact el  
calculo de factorial de n

```
void factorial (int n,unsigned * fact)
{
    *fact=1;
    while (n > 1)
        *fact = *fact * n--;
}
```

## Ejercicio 3.- Enunciado

---

Realiza un programa que utilice una función que tenga la siguiente especificación:

**//cabecera:** entero función cubo (E entero: n)  
**//precondición:**  $n > 0$   
**// postcondición:** devuelve el cubo de n



## Ejercicio 3.- Pseudocódigo

Algoritmo cubo\_valor

Principal

var

entero: num,resp

inicio

repetir

escribir("Introduce numero")

leer(num)

hasta\_que (num > 0)

resp ← cubo(num)

escribir("El cubo de"  
    ,num,"es",resp)

finprincipal

//cabecera: entero funcion cubo  
    (E entero: n)

//precondición:  $n \geq 0$

//postcondición: devuelve el cubo  
    de n

entero funcion cubo (E entero:n)

inicio

devolver (  $n*n*n$  )

finfuncion

finalgoritmo

## Ejercicio 3.- Lenguaje C

```
#include <stdio.h>

int cubo (int); //prototipo

int main(){
    int num,resp;
    do{
        puts("Introduce numero");
        scanf("%d",&num);
    }while (num<0);
    resp=cubo(num);
    printf("El cubo del numero %d
           es %d\n",num,resp);
    return 0;
}
```

```
//cabecera: int cubo(int n)
//precondición: n >= 0
//postcondición: devuelve n al
                  cubo
```

```
int cubo (int n){
    return n*n*n;
}
```

## Ejercicio 4.- Enunciado

---

Realiza un procedimiento que calcule el cubo de un número

La especificación del procedimiento es:

//**cabecera**: procedimiento cubo (E/S entero: pn)

//**precondición**:  $pn > 0$

// **postcondición**: modifica el parámetro actual por el cubo de su valor

## Ejercicio 4.- Pseudocódigo

Algoritmo cubo\_num\_ref

Principal

var

entero: num, copia

inicio

repetir

escribir("Introduce numero")

leer (num)

hasta\_que (num >= 1)

copia ← num

cubo(num)

escribir ("El cubo  
de", copia, "es", num)

finprincipal

//cabecera: procedimiento cubo  
(E/S entero:pn)

//Precondición: pn > 0

//Postcondición: modifica el  
parámetro actual por el cubo  
de su valor

procedimiento cubo (E/S  
entero:pn)

inicio

pn ← pn \* pn \* pn

finprocedimeinto

finalgoritmo

## Ejercicio 4- Lenguaje C

```
#include <stdio.h>
void cubo (int *); //Prototipo
int main(){
    int num,copia;
    do{
        puts("Introduce numero");
        scanf("%d",&num);
    }while (num < 1);
    copia=num;
    cubo(&num);
    printf ("El cubo de %d es
            %d\n",copia,num);
    return 0;}
```

//cabecera: void cubo (int \*pn)  
//Precondición: pn es puntero a  
entero (entero >0)  
//Postcondición: modifica el  
parámetro actual por el cubo  
de su valor

```
void cubo (int *pn){
    *pn=*pn * *pn * *pn;
}
```