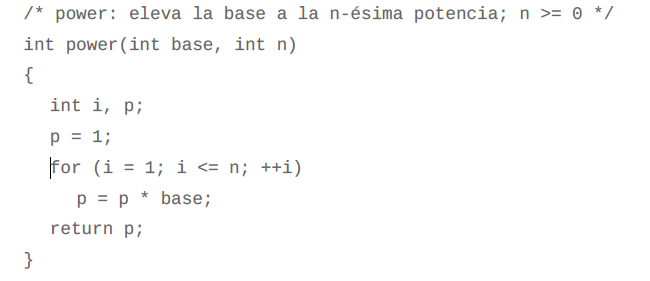
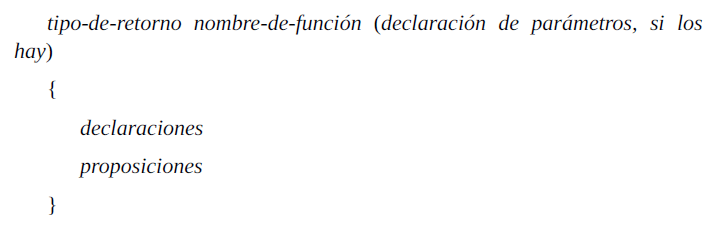
**Capítulos:** 1.7, 1.8, 4.1, 4.2

### 1.7 Funciones

A continuación se muestra la función **power**



Una **definición de función** tiene la forma siguiente:



Los parámetros que emplean las funciones para sus parámetros son locales a la función y son invisibles a cualquier otra. Otras funciones pueden usar los mismo nombres sin que exista problema alguno.

**Parámetro:** Variable nombrada en la lista entre paréntesis de la definición de la función.

**Argumento:** Valor empleado al hacer la llamada de la función.

El valor que retorna una función se regresa por medio de la expresión **return**



Una función no necesita regresar un valor. Además, la función que llama puede ignorar el valor que regresa una función.

### 1.8 Argumentos - llamada por valor

En C, todos los argumentos se pasan por valor. Esto significa que la función que se invoca recibe los valores de sus argumentos en variables temporales y no en las originales.

En C, la **función que se invoca no puede alterar directamente una variable de la función que hace la llamada**; sólo puede modificar su copia privada y temporal.

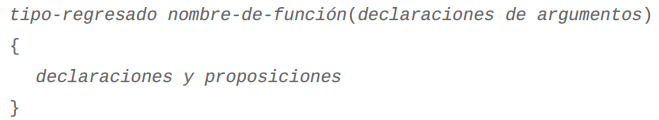
Sin embargo, la llamada por valor es una ventaja. Esto conduce a elaborar programas más compactos con pocas variables extrañas, debido a que los parámetros se tratan en la función invocada como variables locales convenientemente inicializadas.

Cuando sea necesario, es posible hacer que una función modifique una variable dentro de la rutina indicada. La función que llama debe proporcionar la **dirección** de la variable que será cambiada, y la función que se invoca debe declarar el parámetro como **apuntador**.

Cuando el nombre de un **arreglo** se emplea como argumento, el valor que **se pasa a la función es la localización o la dirección del principio del arreglo** (NO HAY COPIA DEL ARREGLO). Al colocarle subíndices, la función puede tener acceso y alterar cualquier elemento del arreglo.

### 4.1 Conceptos básicos de Funciones

Cada definición de función tiene la forma



Si el tipo que regresa la función se omite, se supone int. Es recomendable siempre especificar el tipo de retorno.

### 

### 4.2 Funciones que regresan valores no enteros

Función **atof**

