函数 *scan_pint* 读取从键盘输入的正整数并返回。该函数不接收形参，为了加以说明，在小括号中写入了 **void**。

当然，因为调用方也没有必要赋予实参，所以函数调用运算符（）中是空的。

► 固定程序 **int main(void)** 表示 **main** 函数不包含形参（另外也存在包含形参的 **main** 函数）。

函数返回值的初始化

请大家注意 **main** 函数中声明变量 *nx* 的部分。该变量的初始值是函数 *scan_pint*() 的调用表达式。变量 *nx* 使用函数的返回值（程序执行时从键盘输入的非负的整数值）进行初始化。

► 但是，这种初始化方法只适用于拥有自动存储期的对象（将在 6-3 节为大家介绍），不适用于拥有静态存储期的对象。

作用域

函数 *scan_pint* 和函数 *rev_int* 都包含一个拥有相同标识符（名称）的变量 *tmp*，但它们却是各自独立的不同变量（图 6-10）。

也就是说，函数 *scan_pint* 中的变量 *tmp* 是函数 *scan_pint* 特有的变量，而函数 *rev_int* 中的变量 *tmp* 是函数 *rev_int* 中特有的变量。

赋给变量的标识符，它的名称都有一个通用的范围，称为**作用域**（scope）。

在程序块（复合语句）中声明的变量的名称，只在该程序块中通用，在其他区域都无效。也就是说，变量的名称从变量声明的位置开始，到包含该声明的程序块最后的大括号 } 为止这一区间内通用。这样的作用域称为**块作用域**（block scope）。

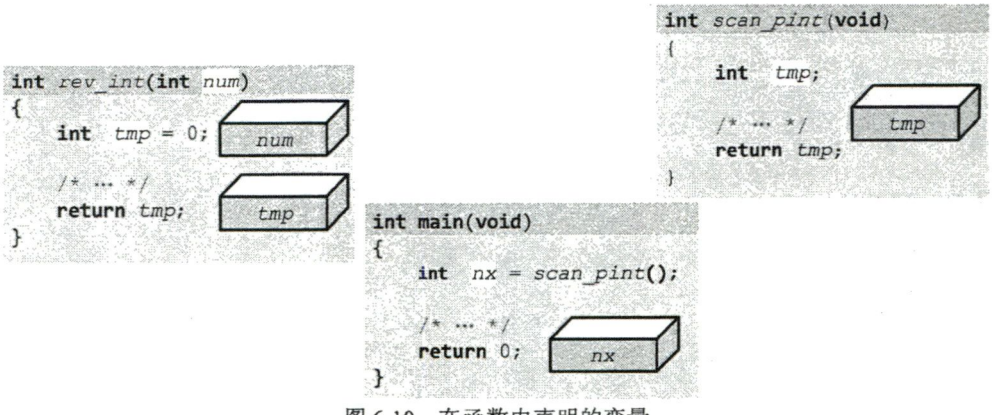

图 6-10 在函数内声明的变量