只有右移操作要求区分有符号和无符号数。这个特性使得补码运算成为实现有符号整数运算的一种比较好的方法的原因之一。

图 3-11 给出了一个执行算术操作的函数示例，以及它的汇编代码。参数 x、y 和 z 初始时分别存放在内存%rdi、%rsi 和%rdx 中。汇编代码指令和 C 源代码行对应很紧密。第 2 行计算 x^y 的值。指令 3 和 4 用 leaq 和移位指令的组合来实现表达式 z* 48。第 5 行计算 t1 和 0x0F0F0F0F 的 AND 值。第 6 行计算最后的减法。由于减法的目的寄存器是%rax，函数会返回这个值。

```
long arith(long x, long y, long z)
{
    long t1 = x ^ y;
    long t2 = z * 48;
    long t3 = t1 & 0x0F0F0F0F;
    long t4 = t2 - t3;
    return t4;
}
```

a）C语言代码

```
    long arith(long x, long y, long z)
    x in %rdi, y in %rsi, z in %rdx
1   arith:
2     xorq    %rsi, %rdi              t1 = x ^ y
3     leaq    (%rdx,%rdx,2), %rax     3*z
4     salq    $4, %rax                t2 = 16 * (3*z) = 48*z
5     andl    $252645135, %edi        t3 = t1 & 0x0F0F0F0F
6     subq    %rdi, %rax              Return t2 - t3
7     ret
```

b）汇编代码

图 3-11　算术运算函数的 C 语言和汇编代码

在图 3-11 的汇编代码中，寄存器%rax 中的值先后对应于程序值 3*z、z* 48 和 t4(作为返回值)。通常，编译器产生的代码中，会用一个寄存器存放多个程序值，还会在寄存器之间传送程序值。

练习题 3.10　下面的函数是图 3-11a 中函数一个变种，其中有些表达式用空格替代：

```
long arith2(long x, long y, long z)
{
    long t1 = _____;
    long t2 = _____;
    long t3 = _____;
    long t4 = _____;
    return t4;
}
```

实现这些表达式的汇编代码如下：

```
long arith2(long x, long y, long z)
x in %rdi, y in %rsi, z in %rdx
```