

 练习题 3.26 函数 fun_a 有如下整体结构：

```
long fun_a(unsigned long x) {
    long val = 0;
    while ( ... ) {
        :
        :
    }
    return ...;
}
```

GCC C 编译器产生如下汇编代码：

```
long fun_a(unsigned long x)
x in %rdi
1  fun_a:
2      movl    $0, %eax
3      jmp     .L5
4  .L6:
5      xorq    %rdi, %rax
6      shrq    %rdi           Shift right by 1
7  .L5:
8      testq   %rdi, %rdi
9      jne     .L6
10     andl    $1, %eax
11     ret
```

逆向工程这段代码的操作，然后完成下面作业：

- 确定这段代码使用的循环翻译方法。
- 根据汇编代码版本填写 C 代码中缺失的部分。
- 用自然语言描述这个函数是计算什么的。

3. for 循环

for 循环的通用形式如下：

```
for (init-expr; test-expr; update-expr)
    body-statement
```

C 语言标准说明(有一个例外，练习题 3.29 中有特别说明)，这样一个循环的行为与下面这段使用 while 循环的代码的行为一样：

```
init-expr;
while (test-expr) {
    body-statement
    update-expr;
}
```

程序首先对初始表达式 *init-expr* 求值，然后进入循环；在循环中它先对测试条件 *test-expr* 求值，如果测试结果为“假”就会退出，否则执行循环体 *body-statement*；最后对更新表达式 *update-expr* 求值。

GCC 为 for 循环产生的代码是 while 循环的两种翻译之一，这取决于优化的等级。也就是，跳转到中间策略会得到如下 goto 代码：

```
init-expr;
goto test;
```