


- A. OP 进行的是什么操作？
 B. 给代码添加注释，解释它是如何工作的。

 练习题 3.21 C 代码开始的形式如下：

```
long test(long x, long y) {
    long val = _____;
    if (_____) {
        if (_____)
            val = _____;
        else
            val = _____;
    } else if (_____)
        val = _____;
    return val;
}
```

GCC 会产生如下汇编代码：

```
long test(long x, long y)
x in %rdi, y in %rsi
test:
    leaq    0(,%rdi,8), %rax
    testq   %rsi, %rsi
    jle     .L2
    movq    %rsi, %rax
    subq    %rdi, %rax
    movq    %rdi, %rdx
    andq    %rsi, %rdx
    cmpq    %rsi, %rdi
    cmovge  %rdx, %rax
    ret
.L2:
    addq    %rsi, %rdi
    cmpq    $-2, %rsi
    cmovle  %rdi, %rax
    ret
```

填补 C 代码中缺失的表达式。

3.6.7 循环

C 语言提供了多种循环结构，即 `do-while`、`while` 和 `for`。汇编中没有相应的指令存在，可以用条件测试和跳转组合起来实现循环的效果。GCC 和其他汇编器产生的循环代码主要基于两种基本的循环模式。我们会循序渐进地研究循环的翻译，从 `do-while` 开始，然后再研究具有更复杂实现的循环，并覆盖这两种模式。

1. `do-while` 循环

`do-while` 语句的通用形式如下：

```
do
    body-statement
while (test-expr);
```

这个循环的效果就是重复执行 `body-statement`，对 `test-expr` 求值，如果求值的结果为非