```
# Homework 3.58
long decode2 (long x, long y, long z) {
    long Step1 = y - z;
    // 将参数 y 减去 z，结果保存在寄存器 %rsi 中
    long Step2 = x * Step1;
    // 将结果乘以参数 x，结果保存在寄存器 %rdi 中
    // movq %rsi, %rax  将参数 y 的值移动到寄存器 %rax 中
    long Step3 = Step2 << 63;
    // salq $63, %rax   左移 63 位，相当于将最高位移到最低位，其他位清零
    long Step4 = Step3 >> 63; // 右移 63 位
    long Step5 = Step4 ^ Step2;
    // xorq %rdi, %rax  将结果与参数 x * (y - z) 异或
    // 结果保存在寄存器 %rax 中
    return Step5;
}
```

3.60

A. 根据汇编代码中的解释，x in %rdi，n in %esi，x 与 n 的寄存器显而易见，而 result 作为输出结果,保存在%rax 寄存器中,mask 掩码作为申请的临时变量，在汇编代码中，保存在%rdx 寄存器当中；

根据按位与、按位或命令，判断条件及跳转语句，可知内部从存在循环，根据代码分析可知：

B. Result = 0，mask = 1；

C. Mask != 0；

D. Mask = mask << n；

E. Result 被修改的过程如下代码所示：

```
long loop(long x, int n) {
    long result = 0;
    long mask;
    for (mask = 1; mask != 0; mask = mask << n ) {
        result = (x & mask) | result ;
    }
    return result;
}
```

3.63

```
long switch_prob (long x, long n) {
    long result = x - 0x3c;
    switch(n) {
        case 0x3c:
```

```
            // 若 %rsi 为 0x0，则跳转到 0x4006f8
        case 0x3e:
            result = x * 8;
            //  若 %rsi 为 0x1，则将 0x0 与 %rdi 的值乘以 8
            break;
        case 0x3f:
            result = x >> 3;
            break;
        case 0x40:
            x = (x << 4) - x;
            break;
        case 0x41:
            x *= x;
            break;
        default:
            result = x + 0x4b;  //  Todo
            break;
    }
    return result;
}
```