

Homework Week7

3.58

```
/* 3.58 */
long decode2(long x, long y, long z)
{
    long res = 0;
    y -= z;
    x *= y;
    res = y;
    res <<= 63;
    res >>= 63;
    res ^= x;
    return res;
    /*
     * Or
     * y = y - z;
     * return ((y << 63) >> 63) ^ (x * y)
     */
}
```

3.60

- A. x in `%rdi`, n in `%esi` and `%ecx`, result in `%rax`, mask in `%rdx`
- B. `mask == 1; result == 0` 因为汇编代码中第 3 行 `movl $1, %edx`, 第 4 行 `movl $0, %eax`
- C. `mask != 0` 因为汇编代码中第 12, 13 行, 当第 12 行测试结果不为 0 时, 第 13 行 `jne` 至循环体, 其中 `testq %rdx %rdx` 的结果即为 `mask` 的值
- D. `mask <= n` 因为汇编代码中第 10 行 `salq %cl, %rdx` 即为将 `mask` 左移 $n \% 256$ 位, 考虑到位移运算会对位移量做模 32 处理, 可忽略掉 `%cl` 和 `%ecx` 的区别
- E. `res |= (x & mask)` 汇编代码中第 7, 8, 9 行得到令 `%rax` 与 `%r8` 做位或运算, 而 `%r8` 是 `%rdi` 和 `%rdx` 的位与运算结果
- F. 代码如下

```
/* 3.60 */
long loop(long x, int n)
{
    long result = 0;
    long mask;
    for (mask = 1; mask != 0; mask <= n) {
        result |= (mask & x);
    }
}
```

```

    return result;
}

```

3.63

从汇编代码中可以看出是先将 x 减去 $0x3c$ /* == 60 */ 后生成的索引为从 0 至 5 的跳转表, 那么 switch 中原本可能的 case 就是从 $60 + 0$ /* == 60 */ 至 $60 + 5$ /* == 65 */ 的
 跳转表中 $0x4006f8 + 0x0$ 与 $0x4006f8 + 0x10$ 的跳转地址相同, 即 case 60 和 case 62 共享代码块
 汇编码中 $0x4005b2$ /* case 64 */ 会穿透至 $0x4005bf$ /* case 65 */, 而后者又会穿透至 $0x4005c3$ /* case 61 */
 故可以按这样的顺序安排他们并省略 break; 语句

```

/* 3.63 */
long switch_prob(long x, long n) {
    long result = x;
    switch(n) {
        case 60:
        case 62:
            result = x * 8;
            break;
        case 63:
            result = x;
            result >>= 3;
            /*
             * Or
             * result = x >> 3;
             */
            break;
        case 64:
            result = x;
            result <<= 4;
            result -= x;
            x = result;
            /*
             * Or
             * x = (x << 4) - x;
             * Or
             * x = x * 15;
             */
        case 65:
            x *= x;
        default:
            result = x + 75; // 75 == 0x4b
            break;
    }
    return result;
}

```