

```

    else
        val = y*z;
    } else if (x > 2)
        val = x*z;
    return val;
}

```

3.19 这道题巩固加强了我们计算预测错误处罚的方法。

A. 可以直接应用公式得到  $T_{MP} = 2 \times (31 - 16) = 30$ 。

B. 当预测错误时，函数会需要大概  $16 + 30 = 46$  个周期。

3.20 这道题提供了研究条件传送使用的机会。

A. 运算符是‘/’。可以看到这是一个通过右移实现除以 2 的 3 次幂的例子(见 2.3.7 节)。在移位  $k=3$  之前，如果被除数是负数的话，必须加上偏移量  $2^k - 1 = 7$ 。

B. 下面是该汇编代码加上注释的一个版本：

```

long arith(long x)
x in %rdi
arith:
    leaq    7(%rdi), %rax    temp = x+7
    testq   %rdi, %rdi      Test x
    cmovns  %rdi, %rax      If x>= 0, temp = x
    sarq    $3, %rax        result = temp >> 3 (= x/8)
    ret

```

这个程序创建一个临时值等于  $x+7$ ，预期  $x$  为负，需要加偏移量时使用。cmovns 指令在当  $x \geq 0$  条件成立时把这个值修改为  $x$ ，然后再移动 3 位，得到  $x/8$ 。

3.21 这个题目类似于练习题 3.18，除了有些条件语句是用条件数据传送实现的。虽然将这段代码装进到原始的 C 代码中看起来有些令人惧怕，但是你会发现它相当严格地遵守了翻译规则。

```

long test(long x, long y) {
    long val = 8*x;
    if (y > 0) {
        if (x < y)
            val = y-x;
        else
            val = x*y;
    } else if (y <= -2)
        val = x*y;
    return val;
}

```

3.22 A. 如果构建一张使用数据类型 int 来计算的阶乘表，得到下面这样的表：

$n$	$n!$	OK?
1	1	Y
2	2	Y
3	6	Y
4	24	Y
5	120	Y
6	720	Y
7	5 040	Y
8	40 320	Y
9	362 880	Y
10	3 628 800	Y
11	39 916 800	Y
12	479 001 600	Y
13	1 932 053 504	N