**3.58**

long decode2(long x, long y, long z) {

y -= z;

x \*= y;

y <<= 63;

y >>= 63;

return y^x;

}

**3.60**

long loop(long x, int n) {

long result = 0;

long mask;

for (mask = 1; mask != 0; mask = mask << (n % 256)){

result |= x & mask;

}

return result;

}

**3.62**

/\*Enumerated type creates set of constants numbered 0 and upward\*/

typedef enum{MODE\_A, MODE\_B, MODE\_C, MODE\_D,MODE\_E} mode\_t;

long switch3(long\* p1, long\* p2, mode\_t action) {

long result = 0;

switch (action) {

case MODE\_A:

result = \*p2;

\*p2 = \*p1;

break;

case MODE\_B:

\*p1 += \*p2;

result = \*p1;

break;

case MODE\_C:

\*p1 = 59;

result = \*p2;

break;

case MODE\_D:

\*p1 = \*p2;

result = 27;

break;

case MODE\_E:

result = 27;

break;

default:

result = 12;

break;

}

return result;

}

**3.66**

注意到第3行汇编代码，result首先增加偏移量3\*n，即第一个矩阵相加元素为A[0] [3\*n]，故NR(n) = 3 \* n;

其次寻找result每次的偏移量，注意到14行，result变量每次加上了%r8内的值，而%r8内的值每次有一次salq $3的操作，即每次乘8，%r8的第一次赋值在第一行，即取地址1 + 4 \* n，故得出NC(n) = 4 \* n + 1;

NR(n) = 3 \* n;

NC(n) = 4 \* n + 1;

**3.68**

由 movslq 8(%rsi), %rax 得知，t存储的位置第二个“8字节”，即array数组宽度至少5字节，至多8字节。得出5<=B<=8;

由 addq 32(%rsi), %rax 得知，s数组至少填满t后的第一个“8字节”即第二个“8字节”的前两个字节，至多填满t后的两个“8字节”，加上补满t后的两个字节。得出7<=A<=10;

由 movq %rax, 184(%rdi) 得知，二维数组x在内存中至多占用184字节，最少占用184-7个字节，即存在对齐的问题。得出184-7<=A\*B\*4<=184;

由三个不等式，解出A, B的唯一解，A=9; B=5.

A=9; B=5.