3.60

A

值	寄存器
x	%rdi　(第 7 行中也把 x 的值存入过%r8 中)
n	%ecx　(作为传入参数也保存在%esi)
result	%rax
mask	%rdx

B

result 初始值为 0, mask 初始值为 1

C

mask != 0;

D

mask <<= (n & 0xFF);

E

result |= (x & mask);

F

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

3.62

```
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;
      action = *p1;
      *p2 = action;
      break;
   case MODE_B:
      result = *p1;
      result += *p2;
      *p1 = result;
      break;
```

```
    case MODE_C:
        *p1 = 59;
        result = *p2;
        break;
    case MODE_D:
        result = *p2;
        *p1 = result;
        result = 27;
        break;
    case MODE_E:
        result = 27;
        break;
    default:
        result = 12;
    }
    return result;
}
```

3.64

&A[i][j][k] = x_A + L * (S * T * i + T * j + k)
其中 X_A 是数组 A 的起始地址,L 是 A 中元素的大小,这道题中 L = 8

```
store_ele:
  leaq (%rsi, %rsi, 2), %rax  # tmp1 = 3 * j
  leaq (%rsi, %rax, 4), %rax  # tmp1 = 13 * j
  movq %rdi, %rsi             #
  salq $6, %rsi               # tmp2 = 64 * i
  addq %rsi, %rdi             # tmp2 = 65 * i
  addq %rax, %rdi             # tmp3 = 65 * i + 13 * j
  addq %rdi, %rdx             # tmp4 = 65 * i + 13 * j + k
  movq A(,%rdx,8), %rax       # tmp1 = x_A + 8 * tmp4
  movq %rax, (%rcx)           # *dest = tmp1
  movl $3640, %eax            # return 3640 (= R * S * T * 8)
  ret
```

所以有 S*T = 65, T = 13, S*T*R = 455;
解得 R = 7, S = 5, T = 13;