3.60. Given:

movl %esi
mo-l src,dest

%rdi	x
%esi	n
%ecx	n
%edx	1
%eax	0
%rdx	0, n*2
%r8	x
%rax	0
%cl	

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

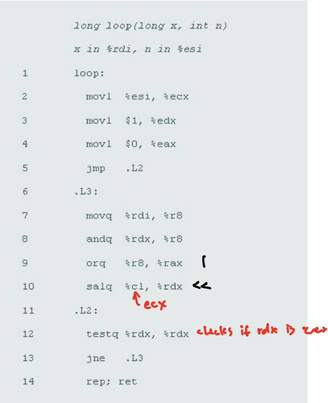
a. n: %esi, %ecx
x: %rdi, %r8
mask: %edx
result: %eax

b. mask =1
result=0

c. mask !=0

d. mask <<2

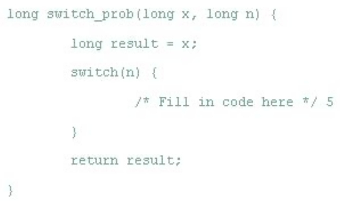
e. result | = mask & x



```
long loop(long x, int n)
x in %rdi, n in %esi
loop:
  movl  %esi, %ecx
  movl  $1, %edx
  movl  $0, %eax
  jmp   .L2
.L3:
  movq  %rdi, %r8
  andq  %rdx, %r8
  orq   %r8, %rax   |
  salq  %cl, %rdx  <<
         ↑ecx
.L2:
  testq %rdx, %rdx checks if rdx is zero
  jne   .L3
  rep; ret
```

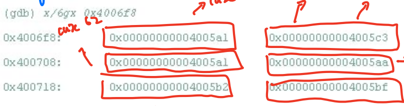
3.63. Jump Table

Function



```
long switch_prob(long x, long n) {
        long result = x;
        switch(n) {
                /* Fill in code here */ 5
        }
        return result;
}
```

Memory Addresses of Jump Table



```
(gdb) x/6gx 0x4006f8
0x4006f8:   0x00000000004005a1   0x00000000004005c3
0x400708:   0x00000000004005a1   0x00000000004005aa
0x400718:   0x00000000004005b2   0x00000000004005bf
```

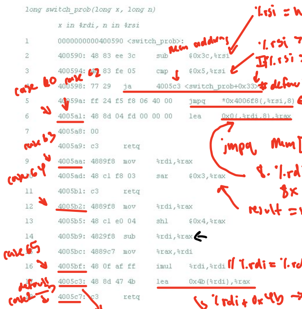
case 62
case 60
case 61
default
case 63
case 65
case 64

*eax <-> %rax

Cases: 0-5
+ default

case 0

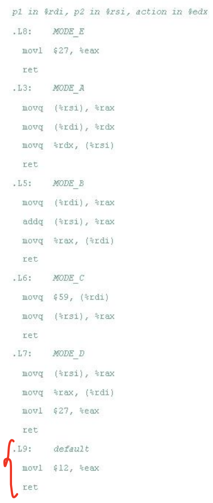
Machine Instructions



```
long switch_prob(long x, long n)
        x in $rdi, n in $rsi
        0000000000400590 <switch_prob>:
        400590: 48 83 ee 3c     sub     $0x3c,%rsi
        400594: 48 83 fe 05     cmp     $0x5,%rsi
        400598: 77 29    ja       4005c3 <switch_prob+0x33>
        40059a: ff 24 f5 f8 06 40 00     jmpq     *0x4006f8(,%rsi,8)
        4005a1: 48 8d 04 fd 00 00 00     lea      0x0(,%rdi,8),%rax
        4005a8: 00
        4005a9: c3      retq
        4005aa: 4889f8  mov     %rdi,%rax
        4005ad: 48 c1 f8 03     sar     $0x3,%rax
        4005b1: c3      retq
        4005b2: 4889f8  mov     %rdi,%rax
        4005b5: 48 c1 e0 04     shl     $0x4,%rax
        4005b9: 4829f8  sub     %rdi,%rax
        4005bc: 4889c7  mov     %rax,%rdi
        4005bf: 48 0f af ff     imul    %rdi,%rdi
        4005c3: 48 8d 47 4b     lea     0x4b(%rdi),%rax
        4005c7: c3      retq
```

%rsi = n-60
%rsi = n-60
mem address
%rsi > 0x5?
If %rsi > 0x5 jump to default
default
case 60 case 62
indirect jumping to 0x04006f8
jmpq Mem[0x4006f8 + 8 %rsi]
8·%rdi+0 -> %rax
8x -> %rax
result = result >> 3
case 63
case 64
case 65
// %rdi = %rdi * %rdi
4b47 8d 48
default case
case 2
%rdi + 0x4b -> %rax
%rdi + 75 -> %rax

Jump Table



```
p1 in $rdi, p2 in $rsi, action in $edx
.L8:    MODE_E
  movl  $27, %eax
  ret
.L3:    MODE_A
  movq  (%rsi), %rax
  movq  (%rdi), %rdx
  movq  %rdx, (%rsi)
  ret
.L5:    MODE_B
  movq  (%rdi), %rax
  addq  (%rsi), %rax
  movq  %rax, (%rdi)
  ret
.L6:    MODE_C
  movq  $59, (%rdi)
  movq  (%rsi), %rax
  ret
.L7:    MODE_D
  movq  (%rsi), %rax
  movq  %rax, (%rdi)
  movl  $27, %eax
  ret
.L9:    default
  movl  $12, %eax
  ret
```

Solution

```
long switch_prob (long x, long n){
        long result=x;
        switch (n){
            case 60:
                result=x*8;
                break;
            case 62:
                result=x*8;
                break;
            case 63:
                result=x;
                result = result >>3;
                break;
            case 64:
                result=x;
                result=result <<4;
                result=result -x;
                x=result;
            case 65:
                x = x*x;
            default:
                result=x+75;
```

Given:

Registers	Value
%rdi	x
%rsi	n
%edx	action
%rax	result

Note:

Sub Src, Dest
Dest = Dest - Src