

Homework 4

—
CS270 Fall 2020

Brennen Green

1 Practice Problems, 3.24, 3.25, 3.28, 3.31 3.32, 3.35

```
(3.24) short loop_while(short a, short b) {
    short result = 0;
    while (a > b) {
        result = b + a ;
        a = a - 1;
    }
    return result;
}
```

```
(3.25) long loop_while(short a, short b) {
    long result = b;
    while (b>0) {
        result *= a;
        b = b-a;
    }
    return result;
}
```

```
(3.28) long test_two(unsigned test x) {
    long val = 0;
    for (int i = 64; i != 0; i--) {
        val = (val << 1) | (x & 0x1);
        x >>= 1;
    }
    return val;
}
```

```

(3.31) void switcher(long a, long b, long c, long *dest) {
    long val;
    switch (a) {
    case 5:
        c = b ^ 15;
    case 0:
        val = c + 112;
        break;
    case 2:
    case 7:
        val = (c + b) << 2;
        break;
    case 4:
        val = a;
        break;
    default:
        val = b;
    }
    *dest = val;
}

```

(3.32)

| Label | PC | Instruction | %rdi | %rsi | %rax | %rsp | *%rsp | Description |
|-------|----------|-------------|------|------|------|---------------|----------|---------------------|
| M1 | 0x400560 | callq | 10 | — | — | 0x7fffffff820 | — | Call first(10) |
| F1 | 0x400548 | lea | 10 | 11 | — | 0x7fffffff818 | 0x400565 | x+1 |
| F2 | 0x40054c | sub | 9 | 11 | — | 0x7fffffff818 | 0x400565 | x-1 |
| F3 | 0x400550 | callq | 9 | 11 | — | 0x7fffffff818 | 0x400565 | Call last(x-1, x+1) |
| L1 | 0x400540 | mov | 9 | 11 | 8 | 0x7fffffff810 | 0x400555 | u |
| L2 | 0x400543 | imul | 9 | 11 | 80 | 0x7fffffff810 | 0x400555 | u * v |
| L3 | 0x400547 | retq | 9 | 11 | 80 | 0x7fffffff810 | 0x400555 | Return |
| F4 | 0x400555 | repz | 9 | 11 | 80 | 0x7fffffff818 | 0x400565 | Return |
| M2 | 0x400565 | mov | 9 | 11 | 80 | 0x7fffffff820 | — | Resume |

(3.35)

```
long rfun(unsigned long x) {  
    if (x == 0) return 0;  
    unsigned long nx = x >> 2;  
    long rv = rfun(nx);  
    return x + rv;  
}
```

2 Homework Problem 3.60

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

- A. x is in %rdi and %r8, n is in %esi and %ecx, result is in %rax, mask is in %rdx
- B. result = 0, mask = 1
- C. mask is not equal to zero
- D. by shifting the lower 8 bits of n to the left
- E. by the bitwise or operation with (x & mask)