

Problem Session 3: Arithmetic and Control Flow in Assembly

September 9, 2020

1. Match each snippet of assembly code on the left with the equivalent C function on the right.

<pre>foo1: movl %edi,%eax sall \$4,%eax subl %edi,%eax ret</pre>	<pre>int choice1(int x){ return (x < 0); } int choice2(int x){ return (x << 31) & 1; }</pre>
<pre>foo2: movl %edi,%eax testl %eax,%eax jge .L4 addl \$15,%eax .L4: sarl \$4,%eax ret</pre>	<pre>int choice3(int x){ return 15 * x; } int choice4(int x){ return (x + 15) / 4 }</pre>
<pre>foo3: movl %edi,%eax shr \$31,%eax ret</pre>	<pre>int choice5(int x){ return x / 16; } int choice6(int x){ return (x >> 31); }</pre>
<pre>foo4: movl %edi,%eax sall \$4,%eax addl %edi,%eax addl %eax,%eax ret</pre>	<pre>int choice7(int x){ return x * 30; } int choice8(int x){ return x * 34; } int choice9(int x){ return a * 18; }</pre>

2. Consider the following assembly code for a C function `looper` and compiled on an x86-64 machine:

```

looper:
    movq    $0, %rax
    movq    $0, %rdx
    jmp     .L2
.L4:
    movq    %rdx, %rcx
    leaq   (%rsi,%rcx,4), %rcx
    cmpq   %rcx, %rax
    jl     .L3
    movq   %rax, %rcx
.L3:
    leaq   1(%rcx), %rax
    addq   $1, %rdx
.L2:
    cmpq   %rdi, %rdx
    jl     .L4
    rep ret

```

(a) For each variable, indicate which register that variable is stored in.

- n: _____
- a: _____
- x: _____
- i: _____

(b) Based on the assembly code, fill in the blanks in the C source code.

```

int looper(int n, int a)
    int x = _____ ;
    for(int i = _____ ; _____ ; i++){
        if (_____ ) {
            x = _____ ;
        } else {
            x = _____ ;
        }
    }
    return x;
}

```