

Week 3: Arithmetic and Conditionals in Assembly

February 6-8, 2023

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

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

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

```
mystery:
    movq    $47, %rax
    cmpq    %rdi, %rsi
    jl     .L1
    addq    %rdi, %rax
    ret
.L1:
    cmpq    %rdi, %rsi
    jg     .L3
    cmpq    %rdi, %rax
    jge    .L2
    ret
.L2:
    addq    %rdi, %rax
    ret
.L3:
    addq    %rsi, %rax
    ret
```

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

- x: _____
- y: _____
- z: _____

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

```
int mystery(int x, int y){
    int z = _____ ;
    if (_____) {
        _____;
    } else if (_____) {
        if (_____) {
            _____;
        } else {
            return 47;
        }
    } else {
        _____;
    }
}
```