

## Section 4

1. What does the following code do?

```
iii)      movl    (%rdi), %eax
          leal    (%eax,%eax,2), %eax
          addl    %eax, %eax
          andl    %esi, %eax
          subl    %esi, %eax
          ret
```

(14au midterm)

2. True or false?

- A logical shift of a 2s-complement number by 3 bits to the right (>> 3) is the same as dividing by 8.
- On x86-64, casting a C float to double has no precision loss.
- A 4-byte integer can be moved into a 32-bit register using a movw instruction.

### [Tutorial Script For Phase 1]

```
gdb bomb
break explode_bomb
break phase_1
break finish_lab    (this function doesn't exist)
run
[input a string]
disas                (shows disassembly of phase 1, also your current place in the program)
help info            (illustrate help command)
info registers       (show the contents of the registers)
q
step                (bomb will explode now unless you magically guessed the right string)
kill                (will hit breakpoint on explode_bomb, don't want it to explode, kill it!!!)
run
[input a string]
stepi
stepi
disas                (show that we are at the function call to compare strings--layout asm is ok too)
x / 10wx $rdi        x /NUM SIZE FORMAT (shows contents of memory at address x)
x / s $rdi           (hey looks like when we interpret the contents as characters...)
x / s $rsi           (let's look at what is in the other register... hey!)
kill
run
[your string for phase 1!]
```

---END---

Misc Tip:

set args (for rerunning with no args)