1. Floating point conversion

m.mmmmeee

$$0x6f = 0110 1111$$

0110 - 1000

2 to the number is the bit that is on so 2³ is the 3rd bit on

2. Vref, voltage, number of bits. Find hex output

Vref = 1v = 1000mv Input = 750 mv Bits = 8	750 / (1000 / 2^8) = 192 = 0xC0
Input / (Vref / 2^8) = Output	

3. Vref, number of bits, digital value. Find voltage range

Vref = 8v = 8000mv Bits = 8	23 / 2^11 * 8v = 11.2 mv
Digital Val = 23	24 / 2^11 * 8v = 11.7mv
Find Range	[11.2mv, 11.7mv)

Vref = 8v Bits = 8	3⁄4 * 8v = 6
Digital Value = 3	4/4 * 8v = 8
	[6,8)

4. Interrupt Trivia

- RETI = set interrupt flag to 1 when return after it pulls 2 bytes off the stack
- Needs special hardware'
- Within an ISR you cannot be interrupted until you turn on the flag
- Timers has the interrupt flag built in
- External interrupts (ex ultrasonic)

5. Given a clock speed, delay. What goes into the timer

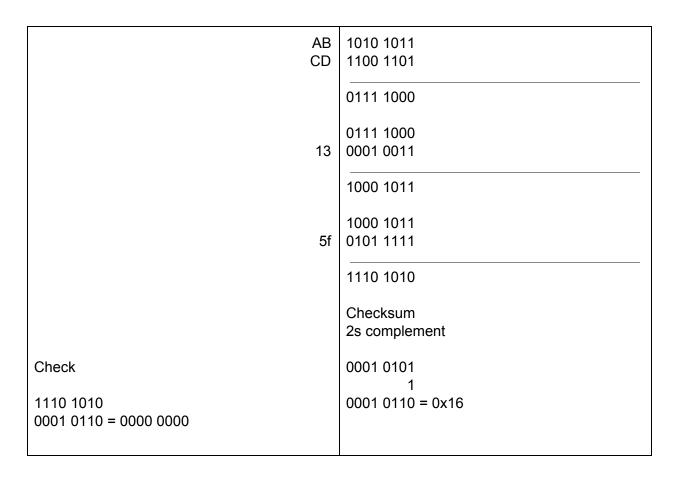
Clock = 5mhz	3x10^-6 / (1 / 5 x 10^6) = 15 clock	
Delay = 3 ms	Tcount = 256 - 15 = 241	

6. Manipulate register bits (LED on/off maybe)

AND	1101 - 1010 0000 - 1111 ← AND
OR	1101 - 1010 0000 - 1111 ← OR
XOR	LDI r16, 1 CLR r17 Loop: XOR r17,r16 toggles the bits

XOR with 3 and 4 bit	ldi r16, 24 EOR r17, r16
swapp r17	

7. Calc CheckSum



8. 2ⁿ bits (Quiz 3)

Unsigned = $0 - (2^10 - 1)$ Signed = $-2^9 - (2^9 - 1)$

9. Add regs tell the SREG Flags

H = 1

C = 1

N = 0

S = 1

V = 1

Z = 0

10. Timer/Counter trivia

```
Timer 0 = 8 bits (256 max)

Timer 1 = 16 bits (1024 max)

Timer 2 = 8 bits (256 max)

CTC = Count To Compare

Load ______ with (timermax - count)
```

11. Assembly instructions to opcode

12. Opcode to assembly

1110 - 0000 - 0010 - 0001	ldi r18, 1
1110 - 0000 - 0011 - 0001	ldi r19, 1
1110 - 0000 - 0100 - 0001	ldi r20, 1
0000 - 1111 - 0011 - 0010	add r19, r18
0010 - 1111 - 0010 - 0011	mov r18, r19
1001 - 0101 - 0100 - 1010 1111 - 0111 - 1110 - 0001	dec r20 brne label (back 4) 111 -1100 000 - 0011 inc 1 then move

13. While / If to assembly

```
while (x > 10) {
    x = x - 2
}

loop:
lds r16, x
cpi r16, 10
brlt out
breq out
dec r16
dec r16
sts x, r16
rjmp loop
out: ret
```

14. Use LPM

```
ldi r30, lo8(MYDATA)
ldi r31, hi8(MYDATA)
clr r17
```

loop: lpm r16, z+

add r17, r16 cpi r16, 0 brne loop (do stuff)

MYDATA: .asciz ("Pizza")