4. ISR question: For system of problem three (Interrupt Controller, 4 bits …) create an Interrupt Service Routine for the following situation. Timer Module is hooked to the most significant bit of the four identified in the question. When the timer service is requested, reset the appropriate flags, increment the value in R11 and send the value to the LEDs. The Interrupt controller address is identified in Problem 3. The Timer Module is located at `0x84460000`, and the LED interface GPIO is located at `0x84480000`. Do not worry about register volatility.

```
.set ISR
.set IRR
.set IER
.set IAR
li r11, 0x0      # r11 initialized to zero outside of routine
.org 0x500       # address for external interrupt service routine
li r20, 0x8      # Pattern to test MSB of the four
lis r21, 0x8446  # Pointer to Timer Module
lis r22, 0x8448  # Pointer to LEDs
li r23, 0x0      # Pattern to set GPIO as output
sfw r23, 4(r22)  # Set GPIO as output.
lis r24, 0x8444  # pointer to interrupt controller
lwz r25, IPR(r24) # check for Interrupt
and r20, r25, r25 # compare
bne end          # if not equal, no interrupt, so leave routine
addi r11, 0x1    # increment value in r11
stw r11, 0(r22)  # Send out to LEDs
sfw r20, IAR(r24) # clear interrupt
end: rfi         # return from interrupt
```

need some more stuff here