3. Interrupt Controller Question: In the table below, identify the registers that need to be initialized, and give values for each. In this interrupt system, there are four interrupt sources, starting in the least significant bit position, and all are to be enabled. Also, the software activation of interrupts is not to be utilized. Assume that you want to assert the appropriate bits to reset any flags that may have remained from an earlier program.

Addr Offset	Register	Bit Pattern
0x00	ISR (Status)	0x 0000 000F ← Status
0x04	IPR (Pending)	read only
0x08	IER (Enable)	0x000 000F ← Enable 4 interrupts
0x0C	IAR (acknowledge)	0x0000 000F ← clear the bits
0x1C	MER	0x0000 0003 ← HIE ME bit

Now, in the space provided below, give instructions that will establish the bit patterns given above as well as to a)set up the vector register (to **0x000A0000**) and b)set up any enabling activity needed to allow interrupts in general. Assume that the interrupt controller has been located at address **0x84440000.**

```
.set ISR, 0x0
.SET IPR, 0x4
.SET IER, 0x8
.SET IAR, 0xC
.SET MER, 0x1C
.SET EVPR, ---

.org 3000
lis r10, 0x8444        # Pointer to interrupt controller
lis r11, 0x000A        # Pointer to vector table
li r13, 0xF            # Pattern for enabling activity
li r14, 0x3            # Pattern for MER
stw r13, ISR(r10)      # Clear previous interrupts
stw r13, IER(r10)      # Enable interrupts
stw r13, IAR(r10)      # Clear interrupts
stw r14, MER(r10)      # Enable Hardware Interrupt
mtspr r11, EVPR        # set up vector register to 0x000A0000
wrteei 1               # set EE bit Enable External Interrupt
```