a. (2 pts) Convert -38 to an 8-bit, two’s complement hex number.

\[ +38 = 0x26, \text{ so } -38 = 0 - (+38) = 0x00 - 0x26 = 0xDA \]

b. (2 pts) What operation and what flag test conditions are used for the comparison “i < k” if i, k are ‘signed int’ variables?

“i < k” is “k > i”, do i – k operation. If “k > i” is true, then i-k result is negative (N=1, V=0). But if overflow occurs, then (N=0, V=1) must also be checked.

c. (2 pts) Write the following in PIC assembly:

\[
\begin{align*}
\text{int } &k, j; \\
&\text{movf } j, w \\
&\text{subwf } k, w \ ; \text{LSB } K = \text{LSB } K - \text{LSB } j \\
&\text{movf } j+1, w \\
&\text{subwfb } k+1, f \ ; \text{MSB } K = \text{MSB } K - \text{MSB } k
\end{align*}
\]

d. (2 pts) In the code below, give the FINAL value of FSR0 and the final value of any changed memory locations after the instruction sequence is executed.

\[
\begin{align*}
\text{movlw} & \text{ FSR0L, 0x59} \\
\text{movlw} & \text{ FSR0H, 0x00} \\
\text{movff} & \text{ PREINC0, 0x058}
\end{align*}
\]

Before the ‘movff’ instruction, FSR0 is initialized to 0x059 by the two movlw instructions.
The ‘preinc0’ increments FSR0 BEFORE it is used, so FSR0++ = 0x5A.
Then the movff becomes:

\[
\text{movff} \ 0x05A, 0x058
\]
so location 0x058 gets the value of location 0x058, or (0x058) = 0xB4

e. What value is pushed on the stack by the ‘call’ instruction in the code below?

\[
\begin{array}{ccc}
\text{Location} & \text{Contents} & \text{Instruction} \\
0x03A4 & \text{EC80 F001} & \text{call } 0x200 \\
0x03A8 & \text{2A40} & \text{incf } 0x040,f
\end{array}
\]

address of next instruction pushed on stack, so 0x03A8 pushed on call/return stack