You may use a calculator and the provided reference materials. If a binary result is required, give
the value in HEX. For any required I2C functionality, use subroutine calls to
\texttt{i2c\_start()}, \texttt{i2c\_rstart()}, \texttt{i2c\_stop}, \texttt{i2c\_put(char byte)}, \texttt{char i2c\_get(char ackbit)}. If you use
\texttt{i2c\_put}, you must pass in as an argument the byte that is to be written to the I2C bus. If you use
\texttt{i2c\_get}, you must pass in as an argument the bit value to be sent back as the acknowledge bit
value.

Part I: (68 pts) You must answer all of these questions.

a. (14 pts) Write a function called \texttt{char adc\_chk (char chan)} that performs a PIC18
conversion on ADC input \texttt{chan} and returns a ‘1’ if
\[ 2.0 \text{ V} \leq \texttt{adc\_val} \leq 3.0 \text{ V}, \]
where \( \texttt{adc\_val} \) is the upper 8-bits of the 10-bit PIC18 ADC
value. The function returns a ‘0’ if \( \texttt{adc\_val} \) is out of this range. Assume \( \texttt{Vref+} = 5 \text{ V}, \)
\( \texttt{Vref-} = 0\text{ V} \) and that the ADC is already configured, with left justification. The \texttt{chan}
value can be 0 to 3, with ‘0’ selecting AN0, ‘1’ selecting AN1, ‘2’ selecting AN2, and ‘3’
selecting AN3. You must delay for 20 \( \mu \text{ s} \) after selecting the channel using \texttt{DelayUs}. 

b. (14 pts) For the hardware setup shown below, write a C function that determines if voltage Va is greater than, less than, or equal to the voltage Vb (use a ‘printf’ statement with a message to this effect). One approach is to start the MAX 517 at 0 V, then increase this voltage in a loop until one (or both) of the comparator outputs becomes a ‘0’ value – this will tell you the relationship between voltages Va and Vb. You could also start the MAX 517 at 5 V, and decrease this voltage in a loop until one or both of the comparator outputs becomes a ‘1’. Assume that the ports RB1 and RB0 are already configured before entering the function.
c. (14 pts) Write a C code fragment that reads three bytes starting at location 0xBF08 in the 24LC515 serial EEPROM into char variables a, b, and c. You must use the I2C function calls listed at the start of the test. Assume the A1, A0 lines of the EEPROM are tied high. When you begin reading bytes, you must read them in the same I2C transaction.
d. (13 pts) Use the PWM module of the PIC18 to generate a square wave with a period of 500 µs and a high pulse width of 100 µs (the low pulse width is 400 µs). Show the calculations that you use to calculate the needed register values. Then write code for `main()` that configures the PWM for this operation. Your `while(1){}` loop should be empty. Use an FOSC value of 10 MHz.

e. (13 pts) Explain EITHER the operation of a 3-bit successive approximation ADC or a 3-bit flash ADC. For both ADCs, use $V_{in} = 2.8\,V$ and $V_{ref} = 4\,V$. If you explain the successive approximation ADC, you have to give the internal VDAC voltage used at each comparison step, and list all steps. If you explain a flash ADC, you have to give the number of comparators and resistors, the output value (1 or 0) of all comparators. For either ADC, you have to give the final 3-bit output code.
Part II: (32 pts) Answer 8 out of the next 10 questions. Cross out the 2 questions that you do not want graded. Each question is worth 4 pts.

1. Draw a diagram that shows a PIC microcontroller connected to two different MAX517 DACs. Label all of the pins.

2. Given an FOSC = 6 MHz, what is the fastest internal clock choice other than the internal A/D oscillator that can be used to drive the ADC clock and not violate the clock period constraint of 1.6 µs?

3. Write a C code fragment that returns the upper 8-bits of the PIC18 ADC result value in the char variable c regardless of whether the ADC is configured as left justified or right justified.
4. If the Vref of the PIC18 is 4.1 V, and I read a 10-bit code of 0x200, what is the input voltage value?

5. How many bit times are there in the I2C transaction to the MAX517 DAC for a conversion? Count the start and stop conditions each as one bit time.

6. Write PIC18 code that configures TIMER2 for prescale of 4, and postscale of 11, internal clock as the clock source, and turns the timer on.

7. What does the least significant bit of the first byte of every I2C transaction signify?
8. Given a 9 bit DAC, and a reference voltage of 4.096 V, what is the expected output voltage change on the DAC output if the input code is changed by 1 Least Significant Bit? Give the answer in **millivolts**.

9. In class and in the notes, we discussed the proper way to WRITE a 16-bit value to timer1 so that it is updated correctly. Show C code that updates TIMER1 with the value from ‘int my_value’ correctly.

10. How many Timer1 tics are in 25 ms assuming an FOSC = 16 MHz and a prescale of 8?