**Question 1.** The first solution technique involves a bridge and instrumentation amplifier.

**Part a)** Ohm’s Law \( R = 1000\,\Omega \), \( V_{in} = 2.5V(1000/2000) - 1.25V = 0V \).

\[
R = 2000\,\Omega, \quad V_{in} = 2.5V(2000/3000) - 1.25V = 0.4167V.
\]

**Part b)** Gain is \( 5V/0.4167V = 12 \).

**Part c)** Resolution = Range/Precision, which is about \((2000\,\Omega - 1000\,\Omega)/1024 = 1\,\Omega\).

**Question 2.** The second solution technique involves a current source and linear amplifier.

**Part a)** The plus terminal is 0V, and because of the negative feedback, the – terminal will also be 0V. Therefore, the current through the 1k\(\,\Omega\) resistor is \( 2.5V/1k\,\Omega = 2.5mA \). Since no current goes in the op amp, this 2.5mA flows across the transducer \( R \).

**Part b)** Using Ohms Law, the voltage range is \(-2.5V < V_{in} < -5V\). There are two valid solutions. If you map –2.5V to 5V and –5V to 0V, then the equation is

\[
V_{out} = 2V_{in} + 10
\]

If you map –2.5V to 0V and –5V to 5V, then the equation is

\[
V_{out} = -2V_{in} - 5
\]

**Part c)** Resolution = Range/Precision, which is about \((2000\,\Omega - 1000\,\Omega)/1024 = 1\,\Omega\).

**Question 3.** The third solution technique involves an astable and period measurement.

**Part a)** Using input capture, the period resolution is 250ns. The precision is 16 bits, so the maximum period that can be measured is \( 65536 \times 0.25\,\mu s = 16384\,\mu s \).

**Part b)** \( P(\mu s) = 0.693 \times C \times (R + 2000) \). At maximum, \( 16384\,\mu s = 0.693 \times C \times (4000) \), so \( C = 5.9\,\mu F \).

**Part c)** Using the above equation, the range of periods will be \( 12266 \) to \( 16354\,\mu s \), with a resolution of \( 0.25\,\mu s \). So, the precision is \( 16352 \) alternatives. The resistance resolution = Range/Precision, which is about \((2000\,\Omega - 1000\,\Omega)/16352 = 0.06\,\Omega \). Another way to get the resolution is to pick two periods that the system can resolve. E.g., it can distinguish \( P=15000\,\mu s \) from \( 15000.25\,\mu s \). The two resistances for these two periods are \( 1668.65\,\Omega \) and \( 1668.71\,\Omega \), so the resolution is \( 0.06\,\Omega \).

**Question 4.** First, you write \$35678 in binary.

\[
\$35678 = 0011\ 0101\ 0110\ 0111\ 1000
\]

Since PPAGE is \( 16K \), you take the bottom 14 bits for the offset and the rest for the page number

Offset = 01 0110 0111 1000
Page = 0011 01

The Page number is used directly, and \$8000 is added to the offset to get the address to access

\[
\text{PPAGE} = \text{0x0D};
\]

\[
\text{data} = *((\text{char} \*) (\text{0x9678}) );
\]
(10) Question 5. The L293 is a good solution, from the data sheet change Vcc2 to +12V, connect 1,2EN to PT0 and ControlA to PT1

(10) Question 6. The worst case data available is the shortest interval, which will be the maximum of [80,100] and the minimum of [200,220].

**DataAvailable** = (100], 200)
The data required interval is the rise of the clock minus setup to rise of the clock plus the hold time

**DataRequired** = (?)Clk-50,?Clk+20)
The worst case data required is the longest interval, which will be the minimum of [170,190] when considering setup and the maximum of [170,190] when considering hold time.

**DataRequired** = (170-50,190+20) = (120,220)
The data will **NOT** be properly stored, because it violates the hold time if the data available ends early at 200, and the ?Clk is late at 190).

(10) Question 7. A fuzzy logic controller

**Part a)** The first slope is 256/(165-80) = 3.0118, which we will approximate as 3. The second slope is 256/(240-197)= 5.9535, which we will approximate as 6.

*Tab:* dc.b 80,3,240,6

or

*Tab:* dc.b $50,$03,$F0,$06

**Part b)** The Fuzzy Logic **OR** is the maximum. The Fuzzy Logic **AND** is the minimum

Healthy*Wise= 200  (maximum of 200,50)

Happy*200 = 100  (minimum of 200,100)

**Part c)** Show the ROM-based data structure uses the 8-bit index, not 16-bit address

**Rules:**

```
dc.b happy,healthy,$FE,good,$FE ; good=good+happy*healthy
dc.b happy,wise,$FE,good,$FE ; good=good+happy*wise
```

dc.b $FF
(10) Question 8. Design an analog circuit
Step one, rewrite with reference chip voltage (none needed).
\[ V_{\text{out}} = 4V_2 + 2V_1 + V_0 \]
Step two, add a ground as an extra input, with a gain such that the sum of the gains is 1.
\[ V_{\text{out}} = 4V_2 + 2V_1 + V_0 - 6V_g \]
Step three, choose a feedback resistor which is a common multiple of 1,2,4,6. \( R_f = 240\, \Omega \).
Step four, select four input resistors to get the desired gains.

(25) Question 9.

```c
int Path[4][4] = { /* Init   InChar  OutChar  Close */
    /* column */ 0, 1, 2, 3 /* */
    /* Init row 0*/ { 0, 1, 1, 1 },
    /* InChar row 1*/ { 0, 1, 1, 1 },
    /* OutChar row 2*/ { 0, 1, 1, 1 },
    /* Close row 3*/ { 1, 0, 0, 0 };
}

Part b)
void SCI_Init(void)
{
    if(Path[State][0]==0) OS_Kill(); // kill if illegal
    State = 0;                        // perform valid Init
    SCIBD = 13;
    SCICR1 = 0;
    SCICR2 = 0x0C;
}

char SCI_InChar(void)
{
    if(Path[State][1]==0) OS_Kill(); // kill if illegal
    State = 1;                        // perform valid InChar
    while((SCISR1 & RDRF) == 0){};
    return(SCIDRL);
}

void SCI_OutChar(char data)
{
    if(Path[State][2]==0) OS_Kill(); // kill if illegal
    State = 2;                        // perform valid OutChar
    while((SCISR1 & TDRE) == 0){};
    SCIDRL = data;
}

void SCI_Close(void)
{
    if(Path[State][3]==0) OS_Kill(); // kill if illegal
    State = 3;                        // perform valid Close
    SCICR2 = 0x00;
}