$$\begin{bmatrix} 20 & 15 & 10 & 45 \\ -3 & -2.249 & 7 & 1.751 \\ 5 & 1 & 3 & 9 \end{bmatrix}$$

a

$R_1 \leftarrow R_1/20$

$$\begin{bmatrix} 1 & .75 & .5 & 2.25 \\ -3 & -2.249 & 7 & 1.751 \\ 5 & 1 & 3 & 9 \end{bmatrix}$$

$R_2 \leftarrow R_2 + 3R_1$
$R_3 \leftarrow R_3 - 5R_1$

$$\begin{bmatrix} 1 & .75 & .5 & 2.25 \\ -3+3(1) & -2.249+3(.75) & 7+3(.5) & 1.751+3(2.25) \\ 5-5(1) & 1-5(.75) & 3-5(.5) & 9-5(2.25) \end{bmatrix} = \begin{bmatrix} 1 & .75 & .5 & 2.25 \\ 0 & .001 & 8.5 & 8.501 \\ 0 & -2.75 & .5 & -2.25 \end{bmatrix}$$

$R_2 \leftarrow R_2 * 1000$

$$\begin{bmatrix} 1 & .75 & .5 & 2.25 \\ 0 & 1 & 8500 & 8501 \\ 0 & -2.75 & .5 & -2.25 \end{bmatrix}$$

$R_1 \leftarrow R_1 - .75R_2$
$R_3 \leftarrow R_3 + 2.75R_2$

$$\begin{bmatrix} 1-.75(0) & .75-.75(1) & .5-.75(8500) & 2.25-.75(8501) \\ 0 & 1 & 8500 & 8501 \\ 0+2.75(0) & -2.75+2.75(1) & .5+2.75(8500) & -2.25+2.75(8501) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -6374.5 & -6373.5 \\ 0 & 1 & 8500 & 8501 \\ 0 & 0 & 23375.5 & 23375.5 \end{bmatrix}$$

$R_3 \leftarrow R_3/23375.5$

$$\begin{bmatrix} 1 & 0 & -6374.5 & -6373.5 \\ 0 & 1 & 8500 & 8501 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$R_1 \leftarrow R_1 + 6374.5R_3$
$R_2 \leftarrow R_2 - 8500R_3$

$$\begin{bmatrix} 1+6374.5(0) & 0+6374.5(0) & -6374.5+6374.5(1) & -6373.5+6374.5(1) \\ 0-8500(0) & 1-8500(0) & 8500-8500(1) & 8501-8500(1) \\ 0 & 0 & 1 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$\therefore$

$x_1 = 1 \quad x_2 = 1 \quad x_3 = 1$

b

$R_1 \leftarrow R_1/20$

$$\begin{bmatrix} 1 & .75 & .5 & 2.25 \\ -3 & -2.249 & 7 & 1.751 \\ 5 & 1 & 3 & 9 \end{bmatrix}$$

$R_2 \leftarrow R_2 + 3R_1$
$R_3 \leftarrow R_3 - 5R_1$

$$\begin{bmatrix} 1 & .75 & .5 & 2.25 \\ -3+3(1) & -2.249+3(.75) & 7+3(.5) & 1.751+3(2.25) \\ 5-5(1) & 1-5(.75) & 3-5(.5) & 9-5(2.25) \end{bmatrix} = \begin{bmatrix} 1 & .75 & .5 & 2.25 \\ 0 & .001 & 8.5 & 8.501 \\ 0 & -2.75 & .5 & -2.25 \end{bmatrix}$$

$R_2 \leftarrow R_2 * 1000$

$$\begin{bmatrix} 1 & .75 & .5 & 2.25 \\ 0 & 1 & 8500 & 8501 \\ 0 & -2.75 & .5 & -2.25 \end{bmatrix}$$

$R_1 \leftarrow R_1 - .75R_2$
$R_3 \leftarrow R_3 + 2.75R_2$

$$\begin{bmatrix} 1-.75(0) & .75-.75(1) & .5-.75(8500) & 2.25-.75(8501) \\ 0 & 1 & 8500 & 8501 \\ 0+2.75(0) & -2.75+2.75(1) & .5+2.75(8500) & -2.25+2.75(8501) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -6374.5 & -6373.5 \\ 0 & 1 & 8500 & 8501 \\ 0 & 0 & 23375.5 & 23375.5 \end{bmatrix}$$

R_3 gets sig-fig'd to 23375

$$\begin{bmatrix} 1 & 0 & -6374.5 & -6373.5 \\ 0 & 1 & 8500 & 8501 \\ 0 & 0 & 23375 & 23375 \end{bmatrix}$$

$R_3 \leftarrow R_3/23375$

$$\begin{bmatrix} 1 & 0 & -6374.5 & -6373.5 \\ 0 & 1 & 8500 & 8501 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$R_1 \leftarrow R_1 + 6374.5R_3$
$R_2 \leftarrow R_2 - 8500R_3$

$$\begin{bmatrix} 1+6374.5(0) & 0+6374.5(0) & -6374.5+6374.5(1) & -6373.5+6374.5(1) \\ 0-8500(0) & 1-8500(0) & 8500-8500(1) & 8501-8500(1) \\ 0 & 0 & 1 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$\therefore$
$x_1 = 1 \quad x_2 = 1 \quad x_3 = 1$

c

Solution wise, it resulted in the same values for $x_1 = x_2, = x_3 = 1$ due to how in step 5 in part A and step 6 in part B the matrix is sitting with
$0x_1 + 0x_2 + 23375.5x_3 = 23375.5$
and
$0x_1 + 0x_2 + 23375x_3 = 23375$
respectively which allows them to divide out to 1.

d

There are a foreseeable challenges when using computing applications, rounding and significant figures is a big one shown in this example, however two other prominent potential issues would be floating point errors and memory/data type allocation.
If an application is using floats vs doubles could result in issues where your decimal may get truncated or rounded, potentially needing even a long double in situations.
A floating point error is similar in idea where you don't have enough "digit size" for a value and the scale of your operands. Since everything breaks down into binary data in a similar format of Scientific notation but with base 2 instead, but when you introduce decimals the storage of it becomes a bit more complex. Take the idea of $1 \div 3$. We would write on paper $\overline{0.3}$ would be the answer, but a computer may try to store all the repeating 3s until the memory it's allocated is full. Results in will depend on the language used, however sometimes still storing that same example could result in wrong data as well. In C, you could get something like the image attached