In [1]:

```
# Define A
A=matrix(QQ,[[1,1,2],[2,4,-3],[3,6,-5]])
show(A)
```

Out[1]:

$$\begin{pmatrix} 1 & 1 & 2 \\ 2 & 4 & -3 \\ 3 & 6 & -5 \end{pmatrix}$$

In [2]:

```
# Define b
b=vector([9,1,0]).column()
show(b)
```

Out[2]:

$$\begin{pmatrix} 9 \\ 1 \\ 0 \end{pmatrix}$$

In [3]:

```
# form the Augmented matrix Ab
Ab=A.augment(b)
show(Ab)
```

Out[3]:

$$\begin{pmatrix} 1 & 1 & 2 & 9 \\ 2 & 4 & -3 & 1 \\ 3 & 6 & -5 & 0 \end{pmatrix}$$

In [4]:

```
# make use of the leading 1 in the first row first column to reduce
# below entries of the same column to be zeros
#
# R2:=R2-2*R1
#
# NOTE THAT SAGE/COCALC counts row from zero rather than 1
# so, in the eyes of CoCalc/Sage, R2:=R2-2*R1 meaning
# take away twice of ROW ZERO (i.e., OUR R1) from ROW 1 (i.e. OUR R2)
# the command is Ab.add_multiple_of_row(i,j,c) for
# adding a scalar multiple c of row j to row i
#
Ab.add_multiple_of_row(1, 0, -2)
show(Ab)
```

Out[4]:

$$\begin{pmatrix} 1 & 1 & 2 & 9 \\ 0 & 2 & -7 & -17 \\ 3 & 6 & -5 & 0 \end{pmatrix}$$

In [5]:

```
#
# R3:=R3-3*R1
#
Ab.add_multiple_of_row(2, 0, -3)
show(Ab)
```

Out[5]:

$$\begin{pmatrix} 1 & 1 & 2 & 9 \\ 0 & 2 & -7 & -17 \\ 0 & 3 & -11 & -27 \end{pmatrix}$$

In [6]:

```
# make entry "2" in the second row second column a leading 1
#
# R2:=(1/2)*R2
#
Ab.rescale_row(1,1/2)
show(Ab)
```

Out[6]:

$$\begin{pmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 3 & -11 & -27 \end{pmatrix}$$

In [7]:

```
# make use of the leading 1 in the second row second column to reduce
# below entry of the same column to be zero
#
# R3:=R3-3*R2
#
Ab.add_multiple_of_row(2, 1, -3)
show(Ab)
```

Out[7]:

$$\begin{pmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 0 & -\frac{1}{2} & -\frac{3}{2} \end{pmatrix}$$

In [8]:

```
# make entry "-1/2" in the third row third column a leading 1
#
# R3:=-2*R3
#
# the command is Ab.rescale_row(i,c) for
# multiplying a scalar multiple c of row i
#
Ab.rescale_row(2,-2)
show(Ab)
```

Out[8]:

$$\begin{pmatrix} 1 & 1 & 2 & 9 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

In [9]:

```
# make use of the leading 1 in the third row third column to reduce
# above entries of the same column to be zeros
#
# R1:=R1-2*R3
#
Ab.add_multiple_of_row(0, 2, -2)
show(Ab)
```

Out[9]:

$$\begin{pmatrix} 1 & 1 & 0 & 3 \\ 0 & 1 & -\frac{7}{2} & -\frac{17}{2} \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

In [10]:

```
#
# R2:=R2+(7/2)*R3
#
Ab.add_multiple_of_row(1, 2, 7/2)
show(Ab)
```

Out[10]:

$$\begin{pmatrix} 1 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

In [11]:

```
# make use of the leading 1 in the second row second column to reduce
# above entry of the same column to be zero
#
# R1:=R1-R2
#
Ab.add_multiple_of_row(0, 1, -1)
show(Ab)
```

Out[11]:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

In [12]:

```
# If you do not want to do it step by step,
# we can also directly compute it in CoCalc
# to get the reduced row-echelon form
# the command is .rref()
#
A=matrix(QQ,[[1,1,2],[2,4,-3],[3,6,-5]])
b=vector([9,1,0]).column()
Ab=A.augment(b)
show(Ab.rref())
```

Out[12]:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{pmatrix}$$

In [0]:

```

```