Name: Amal S. Thundiyil
UID: 2020400066
Batch: IT-D

Scilab No. : 4

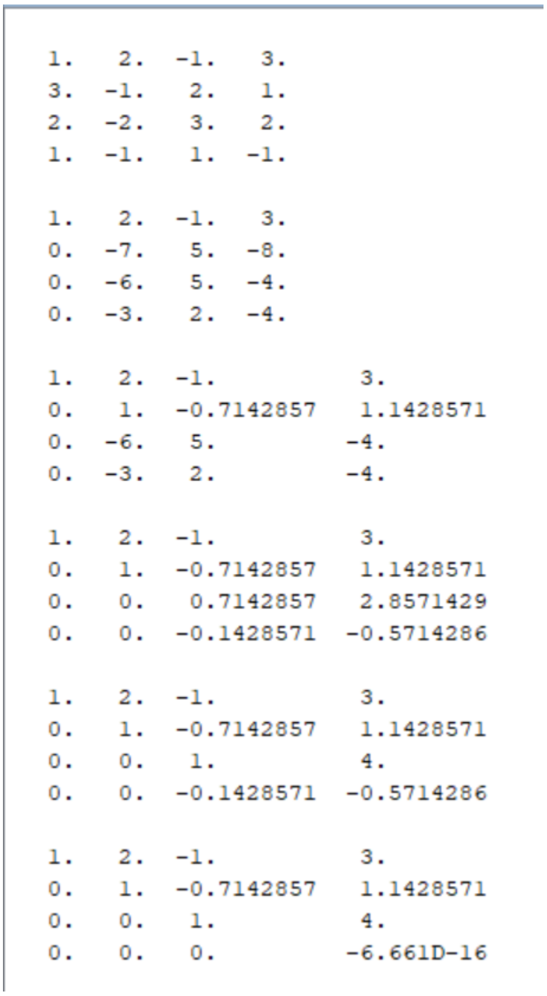
Title: **Row Echelon Form**

Program 1: Write a Scilab code using a for loop to convert a given matrix in row echelon form.

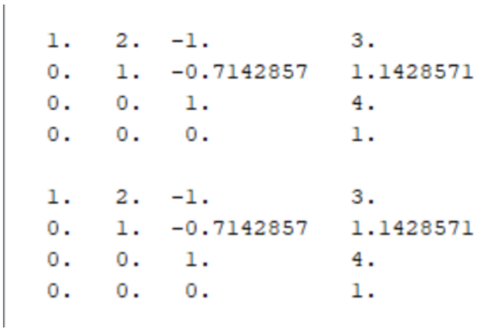
Code:

```
clc
clear all
a = [1 2 -1 3; 3 -1 2 1; 2 -2 3 2; 1 -1 1 -1]
disp(a)
n = 4
for i = 1:n
    if a(i, i) == 0
        a(i,:) = a(i, :)
    else
        a(i, :) = a(i, :)/a(i, i)
    disp(a)
    for j = 1:n-1
        if i+j < n+1
            a(i+j, :) = a(i+j, :) - a(i+j, i)*a(i,:)
        else
        end
    end
end
if a(1, 2) == a(2, 2)
    a(1, :) = a(1, :) - a(2, :)
else
end
disp(a)
end
```

Output :



```
 1.    2.  -1.    3.
 3.  -1.    2.    1.
 2.  -2.    3.    2.
 1.  -1.    1.  -1.

 1.    2.  -1.    3.
 0.  -7.    5.  -8.
 0.  -6.    5.  -4.
 0.  -3.    2.  -4.

 1.    2.  -1.            3.
 0.    1.  -0.7142857    1.1428571
 0.  -6.    5.          -4.
 0.  -3.    2.          -4.

 1.    2.  -1.            3.
 0.    1.  -0.7142857    1.1428571
 0.    0.    0.7142857    2.8571429
 0.    0.  -0.1428571  -0.5714286

 1.    2.  -1.            3.
 0.    1.  -0.7142857    1.1428571
 0.    0.    1.            4.
 0.    0.  -0.1428571  -0.5714286

 1.    2.  -1.            3.
 0.    1.  -0.7142857    1.1428571
 0.    0.    1.            4.
 0.    0.    0.          -6.661D-16
```



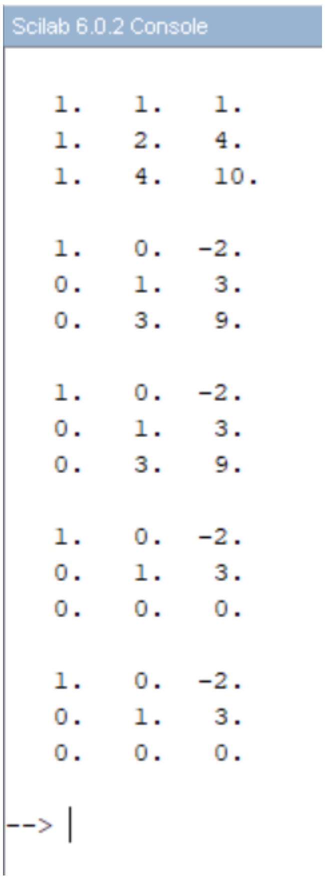
```
 1.    2.  -1.            3.
 0.    1.  -0.7142857    1.1428571
 0.    0.    1.            4.
 0.    0.    0.            1.

 1.    2.  -1.            3.
 0.    1.  -0.7142857    1.1428571
 0.    0.    1.            4.
 0.    0.    0.            1.
```

Program 2 : Write a Scilab code using a for loop to convert a given matrix in row echelon form.

Code :

```
clc
clear all
a  = [1 1 1; 1 2 4; 1 4 10]
disp(a)
n = 3
for i = 1:n
    if a(i, i) == 0
        a(i, :) = a(i, :)
    else
        a(i, :) = a(i, :)/a(i, i)
    disp(a)
    for j = 1:n-1
        if i+j < n+1
            a(i+j, :) = a(i+j, :) - a(i+j, i)*a(i, :)
        else
        end
    end
end
if a(1, 2) == a(2, 2)
    a(1, :) = a(1, :) - a(2, :)
else
end
disp(a)
end
```

Output:



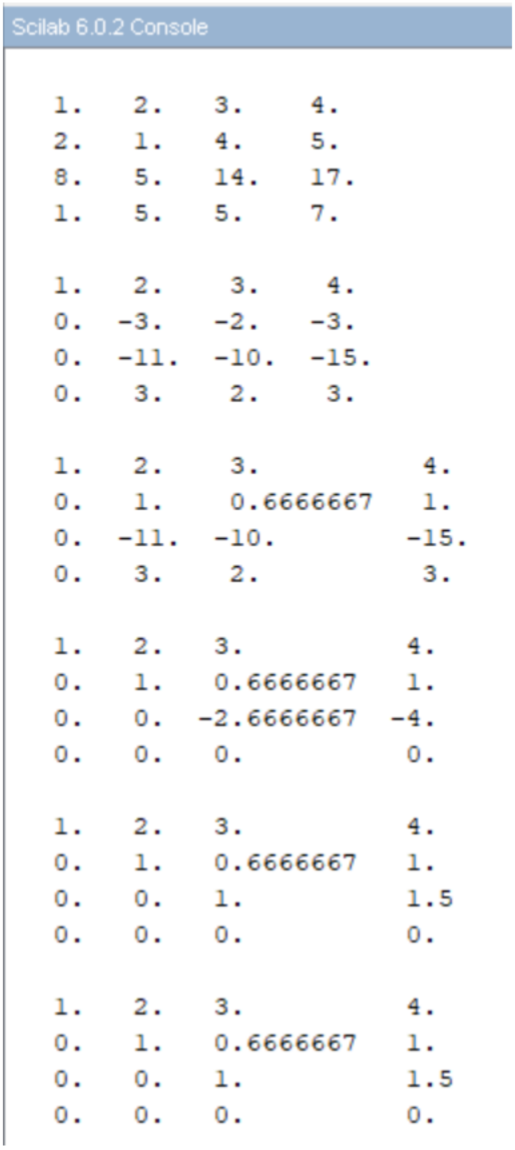
```
   1.   1.   1. 
   1.   2.   4. 
   1.   4.   10.

   1.   0.  -2. 
   0.   1.   3. 
   0.   3.   9. 

   1.   0.  -2. 
   0.   1.   3. 
   0.   3.   9. 

   1.   0.  -2. 
   0.   1.   3. 
   0.   0.   0. 

   1.   0.  -2. 
   0.   1.   3. 
   0.   0.   0. 

--> 
```

Program 3 : Write a Scilab code using a for loop to convert a given matrix in row echelon form.

Code :

```
clc
clear all
a  = [1 2 3 4; 2 1 4 5; 8 5 14 17; 1 5 5 7]
disp(a)
n = 4
for i = 1:n
    if a(i, i) == 0
        a(i, :) = a(i, :)
    else
        a(i, :) = a(i, :)/a(i, i)
    disp(a)
    for j = 1:n-1
        if i+j < n+1
            a(i+j, :) = a(i+j, :) - a(i+j, i)*a(i, :)
        else
        end
    end
end
if a(1, 2) == a(2, 2)
    a(1, :) = a(1, :) - a(2, :)
else
end
disp(a)
end
```

Output:



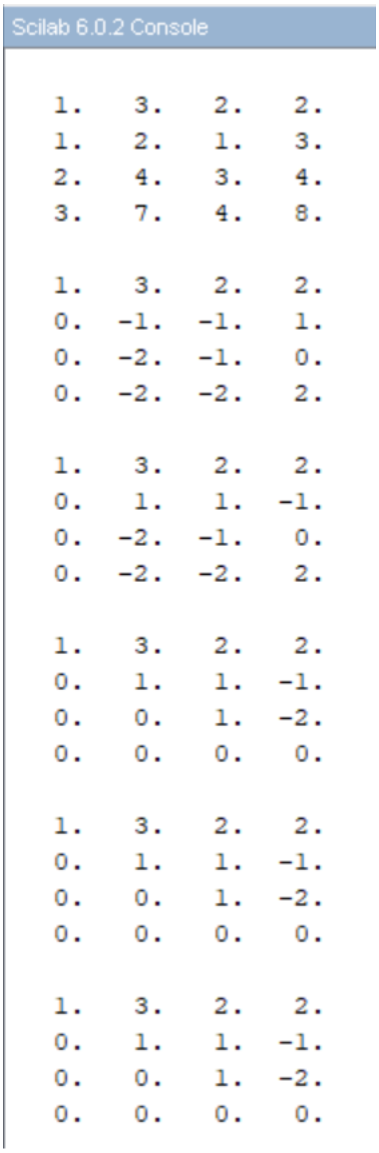
```
   1.   2.   3.    4.
   2.   1.   4.    5.
   8.   5.   14.   17.
   1.   5.   5.    7.

   1.   2.    3.    4.
   0.  -3.   -2.   -3.
   0.  -11.  -10.  -15.
   0.   3.    2.    3.

   1.   2.    3.          4.
   0.   1.    0.6666667   1.
   0.  -11.  -10.        -15.
   0.   3.    2.          3.

   1.   2.   3.          4.
   0.   1.   0.6666667   1.
   0.   0.  -2.6666667  -4.
   0.   0.   0.          0.

   1.   2.   3.          4.
   0.   1.   0.6666667   1.
   0.   0.   1.          1.5
   0.   0.   0.          0.

   1.   2.   3.          4.
   0.   1.   0.6666667   1.
   0.   0.   1.          1.5
   0.   0.   0.          0.
```

Program 4: Write a Scilab code using a for loop to convert a given matrix in row echelon form.

Code :

```
clc
clear all
a  = [1 3 2 2; 1 2 1 3; 2 4 3 4; 3 7 4 8]
n = 4
for i = 1:n
    if a(i, i) == 0
        a(i, :) = a(i, :)
    else
        a(i, :) = a(i, :)/a(i, i)
    disp(a)
    for j = 1:n-1
        if i+j < n+1
            a(i+j, :) = a(i+j, :) - a(i+j, i)*a(i, :)
        else
        end
    end
end
if a(1, 2) == a(2, 2)
    a(1, :) = a(1, :) - a(2, :)
else
end
disp(a)
end
```

Output:



Scilab 6.0.2 Console

```
   1.   3.   2.   2.
   1.   2.   1.   3.
   2.   4.   3.   4.
   3.   7.   4.   8.

   1.   3.   2.   2.
   0.  -1.  -1.   1.
   0.  -2.  -1.   0.
   0.  -2.  -2.   2.

   1.   3.   2.   2.
   0.   1.   1.  -1.
   0.  -2.  -1.   0.
   0.  -2.  -2.   2.

   1.   3.   2.   2.
   0.   1.   1.  -1.
   0.   0.   1.  -2.
   0.   0.   0.   0.

   1.   3.   2.   2.
   0.   1.   1.  -1.
   0.   0.   1.  -2.
   0.   0.   0.   0.

   1.   3.   2.   2.
   0.   1.   1.  -1.
   0.   0.   1.  -2.
   0.   0.   0.   0.
```