

```

%William Archbold
%Assignment 3 for Linear Algebra
%06 Nov 2019
for file = 1:2
sysMat = load(['sysmat',num2str(file),'.txt']);
A = reshape(sysMat(2:end), sysMat(1),[]); %reshape by 1st element number in
sysMat, which is 3
A = A.'; %transpose matrix

sysVec = load(['prodvec',num2str(file),'.txt']);
V = sysVec(2:end); %get rid of the size parameter
V = V.'; %turn from row vector to column vector


                                %tilde skips argument
                                %ind stores index of max
                                %indexing starts at 1 and goes down columns
                                %first
                                %: = all rows of A and first column
                                %concatenate V to A
A = [A,V];
disp(['Matrix',num2str(file),':']);
disp("The System Matrix:");
disp(A);
%A = A([2,1,3],:) %move second row to first row

for k = 1:size(A,2)-1 %from k = 1 to size of columns (1 means rows, 2
means columns)
    %find value in column with highest abs value
    [~, ind] = max(abs(A(k:end,k))); %k:end is k to rest of all rows
    ind = ind+k-1; %k moves ind out of zeroized columns
    A1 = A; %temp matrix to store old top row
    A(k,:)= A(ind,:); %automate moving ind to first row
    A(k,:) = A(k,:)./norm(A(k,:)); %divide components of row by the norm to
normalize row
                                %needed that line because matrix 2
                                %numbers were getting way too big
    A(ind,:) = A1(k,:); %place stored first column into moved row to complete
swap

for loop = 2+k-1:size(A,1) %A,1 size of rows A,2 size of columns
    A(loop,:) = A(k,:)*A(loop,k) + A(loop,:)*-A(k,k);

    %below is example from which above was automated
    %A(2,:) = A(1,:)*A(2,1) + A(2,:)*-A(1,1) %A 2nd row equals..
    %A(3,:) = A(1,:)*A(3,1) + A(3,:)*-A(1,1) %A 3rd row equals

end
    if k~=size(A,2)-1 %don't display for last column of matrix
        disp(["Column" num2str(k)]);
        disp(A);
    end
end

```

```

[m,n] = size(A);    %get dimensions
x = zeros(m,1); %zeroes preallocate an array of x's for every row
for k1 = m:-1:1    %decrementing loop by finding last x first
    summation=0;    %summation adds all the coefficients
    for k2 = 2:m    %for every row on left side of Ax = v
        summation = summation + A(k1,k2)*x(k2);
    end
    x(k1) = (A(k1,n)-summation)/A(k1,k1); %A(k1,k1) represents all diagonal
coefficients

end
disp("Solution Vector: ");
disp(x);

end

```