

```
%Gaussian elimination with partial pivoting
```

```
n = 10;
%{
% Case a)
%identity = eye(n)
A = zeros(n);
b = [4:0.7:(0.7*n+3.3)]';
% fill in a
for i = 1:n
    for j = 1:n
        if i == j
            A(i, j) = 8;
        elseif (i == j-1) || (i == j+1)
            A(i, j) = 2;
        end
    end
end
end
%}
```

```
% Case b)
A = zeros(n);
b = zeros(n, 1);
% fill in a
for i = 1:n
    for j = 1:n
        A(i, j) = 3/(5*(i+j+1));
    end
end
% fill in b
for i = 1:n
    if mod(i, 2) == 1
        b(i) = 0;
    else
        b(i) = 5/(4*i);
    end
end
```

```
A;
b;
```

```
%Step 1 - Put into Triangluar Form
% merge A and b
triangle = [A b];
% for each row, put it into specail function
for i = 1:n-1
```

```

    [maxValue maxValueIndex] = max(abs(triangle(i:end, i)));
    maxValueIndex = maxValueIndex + (i-1);

    tempRow = triangle(maxValueIndex , :);
    triangle(maxValueIndex , :) = triangle(i, :);
    triangle(i, :) = tempRow;
    for j = i:n-1
        row1 = triangle(i, i:end);
        row2 = triangle(j+1, i:end);
        triangle(j+1, i:end) = zeroFristElement(row1, row2);
    end
end

```

```

%Step 2- Solve by back substitution
x = zeros(n,1)

```

```

x = 10x1
    0
    0
    0
    0
    0
    0
    0
    0
    0
    0

```

```

for i = n:-1:1
    numerator = triangle(i, end) - sumPreviousElements (triangle(i, i:end-1)', x(i:end-1));
    x(i) = numerator/triangle(i, i);
end
x

```

```

x = 10x1
1013 x
   -0.0001
    0.0051
   -0.0646
    0.4130
   -1.5209
    3.4222
   -4.7826
    4.0507
   -1.9038
    0.3810

```

```

matlabX = mldivide(A, b)

```

```

matlabX = 10x1
1013 x
   -0.0001
    0.0051
   -0.0646
    0.4129
   -1.5204

```

```
3.4212
-4.7812
4.0495
-1.9033
0.3809
```

```
function [newRow] = zeroFristElement (row1, row2)
    rowMultiplier = row2(1,1) / row1(1,1);
    newRow = row2' - rowMultiplier * row1';
    newRow = newRow';
end

function [newSum] = sumPreviousElements (colA, colX)
    newVec = colA .* colX;
    newSum = sum(newVec, 'all');
end
```

```
% find Ax = b, without A/b

%r = (A*x)-b
%delX = mldivide(A,r)
%newX = x - delX
%newR = (A*newX)-b
%norm(r)
%norm(newR)
```