

```
A=[1 1 -1;0 1 3; -1 0 2]
```

```
A = 3x3
     1     1    -1
     0     1     3
    -1     0     2
```

```
b= [9;3;2]
```

```
b = 3x1
     9
     3
     2
```

```
x=gaussElimination(A,b)
```

```
x = 3x1
    -10
     15
     -4
```

```
function x=gaussElimination(A,b)
    %%%%%%%%%Verifications
    [m,n] = size(A);
    if m~=n, error('La matriz A no es cuadrada'), end
    if length(b) ~= m, error('b no tiene el tamaño de la matriz A'), end
    %%%%%%%%%Forward elimination
    for i =1:n-1
        [X,c] = max(abs(A(i:m,i))); %Obtenemos ubicación del máximo
        col = i+(c-1); %Obtenemos columna de cambio
        if col~=i
            A([col,i],:) = A([i,col],:); %intercambiar renglones
            b([col,i],:) = b([i,col],:);
        end
        %Realizar operaciones por renglón
        for h=i+1:m
            mul = A(h,i)/A(i,i);
            subrenglon = i:n;
            A(h,subrenglon) = A(h,subrenglon)- mul*A(i,subrenglon);
            b(h) = b(h)- mul*b(i);
        end
    end

    %%%%%%%%%Check matrix
    for i=1:n
        if A(i,i)==0, error('No hay solución única'),end
    end

    %%%%%%%%%Back substitution
    x = zeros(m,1);
    %Empezando con el último valor
    x(m) = b(m)/A(m,n);
    for i = m-1:-1:1
        %Comienza la sustitución de valores con respecto al último punto
        %obtenido de abajo hacia arriba
    end
end
```

```
        x(i) = (b(i)-A(i,i+1:n)*x(i+1:n));  
    end  
end
```