```
format long e
```

Question 3 (a)

```
A = [1e-20, 1; 1, 1];
[L, U] = genp(A);
q3a_ans= A - L * U
```

```
q3a_ans =
     0     0
     0     1
```

Question 3 (b)

```
b = [1; 0];
xc = solve(A, b);
x = A \ b;
q3b_ans = norm(xc - x) / norm(x)
```

```
q3b_ans =
     7.071067811865475e-01
```

We can conclude that GENP may not give a correct solution when the pivot value is very small even though the function does not exit.

In the first step itself, dividing by 1e-20 makes the last row quite large.

Question 4

```
L_norm = zeros(1, 10);
U_norm = zeros(1, 10);
for i = 1:10
    A = randn(10);
    [L_hat, U_hat, ~] = gepp(A);
    [L, U, ~] = lu(A);
    L_norm(i) = norm(L_hat - L) / norm(L);
    U_norm(i) = norm(U_hat - U) / norm(U);
end
L_norm
```

```
L_norm =
     1.948829867736452e-16     6.768878265393872e-16     1.371431919900664e-16     2.214909043237717e-16
```

```
U_norm
```

```
U_norm =
     2.734446865268963e-16     3.136616222272667e-16     2.045353789745362e-16     2.152617336200437e-16
```

Question 5 (a)

```
x_norm = zeros(1, 10);
for i = 1:10
    A = randn(10);
    b = randn(10, 1);
```

```
    x = geppsolve(A, b);
    x_hat = A \ b;
    x_norm(i) = norm(x_hat - x) / norm(x);
end
x_norm
```

```
x_norm =
     4.179645902363428e-16     3.331549456635863e-14     4.902262520206317e-16     7.687834629317253e-16
```

Question 5 (b)

```
A = [1e-20, 1; 1, 1];
[L, U, p, ~] = gepp(A);
q5b_ans_1= A(p, :) - L * U
```

```
q5b_ans_1 =
     0     0
     0     0
```

```
b = [1; 0];
xc = geppsolve(A, b);
x = A \ b;
q5b_ans_2 = norm(xc - x) / norm(x)
```

```
q5b_ans_2 =
     0
```

As we can see, the answer obtained by GEPP is far more accurate than the answer obtained by GENP.

Question 6

```
A = randn(10);
determinant = mydet(A)
```

```
determinant =
   -8.621549577366590e+00
```

```
determinant_matlab = det(A)
```

```
determinant_matlab =
   -8.621549577367128e+00
```

Functions

Question 1 (a)

```
function x = colbackward(U, b)
    [n, ~] = size(U);
    x = zeros(n, 1);
    for i = n:-1:1
        x(i) = b(i) / U(i,i);
        b(1:i-1) = b(1:i-1) - U(1:i-1,i) * x(i);
    end
end
```

Question 1 (b)

```
function x = rowforward(L, b)
    x = zeros(size(b));
    n = length(x);
    x(1) = b(1) / L(1, 1);
    for i = 2:n
        sum = 0;
        for j = 1:(i - 1)
            sum = sum + L(i, j) * x(j);
        end
        x(i) = (b(i) - sum) / L(i, i);
    end
end
```

what if L(i,i) == 0?

Question 2

```
function [L, U] = genp(A)
    sz = size(A);
    n = sz(1);
    for k = 1:(n - 1)
        if A(k, k) ~= 0
            A((k + 1):n, k) = A((k + 1):n, k) / A(k, k);
        else
            exit('zero pivot encountered');
        end
        A((k + 1):n, (k + 1):n) = A((k + 1):n, (k + 1):n) - A((k + 1):n, k)
* A(k, (k + 1):n);
    end
    L = tril(A, -1) + eye(size(A));
    U = triu(A);
end
```

Question 3 (b)

```
function x = solve(A, b)
    [L, U] = genp(A);
    y = rowforward(L, b);
    x = colbackward(U, y);
end
```

Question 4

```
function [L, U, p, sign] = gepp(A)
    [n, ~] = size(A);
    p = (1: n);
    sign = 1;
    for k = 1: n - 1
        [~, idx] = max(abs(A(k: end, k)));
        idx = idx + k - 1;
```

```
        if idx ~= k
            sign = -1 * sign;
            permutation = [idx, k];
            A(permutation, :) = A(permutation([2, 1]), :);
            p(permutation) = p(permutation([2, 1]));
        end

        if A(k, k) ~= 0
            A(k + 1: n, k) = A(k + 1: n, k) / A(k, k);
        end

        A(k + 1: n, k + 1: n) = A(k + 1: n, k + 1: n) - A(k + 1: n, k) *
A(k, k + 1: n);
    end

    L = eye(n) + tril(A, -1);
    U = triu(A);
end
```

Question 5 (a)

```
function x = geppsolve(A, b)
    [L, U, p, ~] = gepp(A);
    y = rowforward(L, b(p, :));
    x = colbackward(U, y);
end
```

Question 6

```
function d = mydet(A)
    [~, U, ~, sign] = gepp(A);
    [n, ~] = size(A);
    d = sign;

    for i = 1: n
        d = d * U(i, i);
    end
end
```