Question2: **2.1 Computer Problems 1.**

Put together the code fragments in this section to create a Matlab program for "naive'' Gaussian elimination (meaning no row exchanges allowed). Use it to solve the systems of Exercise 2.

Applying naïve Gaussian elimination in appendix, we have solved the following equations:

(a)

$$\begin{cases} 2x - 2y - z = -2 \\ 4x + y - 2z = 1 \\ -2x + y - z = -3 \end{cases} \quad \textit{solves to} \quad \begin{cases} x = 1 \\ y = 1 \\ z = 2 \end{cases}$$

(b)

$$\begin{cases} x + 2y - z = 2 \\ 3y + z = 4 \\ 2x - y + z = 2 \end{cases} \quad \textit{solves to} \quad \begin{cases} x = 1 \\ y = 1 \\ z = 1 \end{cases}$$

(c)

$$\begin{cases} 2x + y - 4z = -7 \\ x - y + z = -2 \\ -x + 3y - 2z = 6 \end{cases} \quad \textit{solves to} \quad \begin{cases} x = -1 \\ y = 3 \\ z = 2 \end{cases}$$

```
% print Gaussian.m file:
function y=gaussian(a,b,n)
for j =1:n-1 % j is column number
    if abs(a(j,j))<eps; error('zero pivot encountered'); end
    for i = j+1:n
        % eliminate entry a(i,j)
        mult = a(i,j)/a(j,j);
        for k = j+1:n
            a(i,k) = a(i,k) - mult*a(j,k);
        end
        b(i) = b(i) - mult*b(j);
    end
end

for i = n : -1 : 1
    for j = i+1 : n
        b(i) = b(i) - a(i,j)*x(j);
    end
    x(i) = b(i)/a(i,i);
end
y=x;
end
```

Question3: 2.1 Computer Problems 2.

Let H denote the n × n Hilbert matrix, whose (i, j) entry is 1/(i + j − 1). Use the Matlab program from Computer Problem 1 to solve Hx = b, where b is the vector of all ones, for (a) n = 2 (b) n = 5 (c) n = 10.

(a) n = 2, solution will be (x1,x2)=(-2.0000, 6.0000)

(b) n = 5, solution will be (x1,x2,x3,x4,x5)=
(5.0000, -120.0000, 630.0000, -1120.0000, 630.0000)

(c) n = 10, solution will be (x1,x2,x3,x4,x5,x6,x7,x8,x9,x10)=
(-9.9974, 989.7719, -23755.1338, 240195.71429, -1261048.5972, 3783198.5011, -6725765.4896, 7000357.2379, -3937735.4176, 923673.4085)

Note: For simplicity, I keep four decimals in the results.

```
% print hw2q3.m file:
n=2; % n=5; n=10 in other cases
for i = 1:n
    for j = 1:n
        h(i,j) = 1/(i+j-1);
    end
end
b=ones(1,n);
sol = gaussian(h,b,n)
```

Question5: **2.2 Computer Problems 1.**

Use the code fragments for Gaussian elimination in the previous section to write a Matlab script to take a matrix A as input and output L and U. No row exchanges are allowed—the program should be designed to shut down if it encounters a zero pivot. Check your program by factoring the matrices in Exercise 2.

Apply A = L* U factorization for matrices below, we get

(a) $\begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 3 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$

(b) $\begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0.5 & 0.5 & 1 \end{bmatrix} * \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 2 \end{bmatrix}$

(c) $\begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 1 & 3 & 4 & 4 \\ 0 & 2 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & -1 & 1 & 2 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 \end{bmatrix}$

```
% print hw2q5.m file:
a=[3,1,2,;6,3,4;3,1,5]; % b=[4,2,0;4,4,2;2,2,3]; c=[1,-
1,1,2;0,2,1,0;1,3,4,4;0,2,1,-1];
LUfactor(a)

function [L,U]= LUfactor(a)
[n,n]=size(a);
L=zeros(n,n);
U=zeros(n,n);
for i=1:n
    if abs(a(i,i))<=0;error('zero pivot encountered'); end
    for k=1:i-1
        L(i,k)=a(i,k);
        for j=1:k-1
            L(i,k)=L(i,k)-L(i,j)*U(j,k);
        end
        L(i,k)=L(i,k)/U(k,k);
    end

    for k=1:n
        U(i,k)=a(i,k);
        for j=1:i-1
            U(i,k)=U(i,k)-L(i,j)*U(j,k);
        end
    end
L(i,i)=1;
end
```

Question8: 2.3 Computer Problems 1.

For the n × n matrix with entries Aij = 5/(i + 2j − 1), set x = [1, . . . ,1]T and b = Ax. Use the Matlab program from Computer Problem 2.1.1 or Matlab's backslash command to compute xc, the double precision computed solution. Find the infinity norm of the forward error and the error magnification factor of the problem Ax = b, and compare it with the condition number of A: (a) n = 6 (b) n = 10.

(a) For n=6, infinity norm of forward error = 8.6652e-11
 the error magnification factor= 5.9756e+05
 condition number of A is 7.0342e+07
 so error magnification factor is smaller than condition number.

(b) For n=10, infinity norm of forward error = 9.1014e-04
 the error magnification factor= 3.7517e+12
 condition number of A is 1.3137e+14
 so error magnification factor is smaller than condition number.

```
% print hw2q8.m file:
n=6; % n =10
for i = 1:n
    for j = 1:n
        a(i,j) = 5/(i+2*j-1);
    end
end
x=ones(n,1);
b=a*x;
xc=a\b;
rel_f_error=max(abs(x-xc))/max(abs(x)); %5.35 ¡Á 10?10
rel_b_error=max(abs(b-a*xc))/max(abs(b));
err_mag_fac=rel_f_error/rel_b_error;
cond(a,inf)
```