Ejercicio 1

a)

```
function[Q,R] = qr1(A)
    % Tamaño de la matriz A
    [m, n] = size(A); %m filas y n columnas

    % Inicializar Q con ceros y tamaño como A
    Q = zeros(m, n);

    % Inicializar R de n x n
    R = zeros(n, n);

    %Ciclo que recorre las columnas
    for j = 1:n
        % Tomar la columna j de A y la asigna a v_j
        vj = A(:, j);

        % Bucle interno para calcular las proyecciones
        % Note que cuando j=1 el ciclo va de 1:0, es decir no se ejecuta.
        for i = 1:j-1

            % Calcular r_ij, es producto interno
            R(i, j) = Q(:, i)' * A(:, j);

            % Modifica v_j restando la proyección de a_j en q_i
            vj = vj - R(i, j) * Q(:, i);
        end

        % Paso 3: calcula r_jj como la norma de v_j y normaliza para obtener q_j
        R(j, j) = norm(vj);
        Q(:, j) = vj / R(j, j);
    end
end
```

b)

```
function[Q,R] = qr2(A)
    % Tamaño de la matriz A
    [m, n] = size(A); %m filas y n columnas

    % Inicializar Q con ceros y tamaño como A
    Q = zeros(m, n);

    % Inicializar R de n x n
    R = zeros(n, n);
```

```
    % Definir la matriz A igual a una V
    V = A;

    % Ciclo for que recorre columnas de v
    for i = 1:n

        % Llenar el valor r_ii
        R(i, i) = norm(V(:, i));

        % Llenar la columna q_i
        Q(:,i) = V(:, i)/R(i, i);

        % Ciclo for interno de las proyecciones
        for j = i+1:n

            % Llenar la fila de la triangular
            R(i, j) = Q(:, i)' * V(:, j);

            % Modifica v_j restando la proyección de a_j en q_i
            V(:, j) = V(:,j) - R(i, j) * Q(:, i);
        end
    end
end
```

c)

```
function [Q, R] = qr3(A)
    [m, n] = size(A);        % Tamaño de la matriz A
    R = A;                   % Inicializamos R como una copia de A
    M = eye(m);              % Matriz identidad de m x m para construir Q

    for j = 1:n
        % Selecciona el vector x desde la fila j hasta m de la columna j de R
        x = R(j:m, j);

        % Construye el vector de Householder vj
        e1 = zeros(length(x), 1);
        e1(1) = 1;
        vj = sign(x(1)) * norm(x) * e1 + x;
        vj = vj / norm(vj);  % Normaliza vj

        % Actualiza R aplicando la transformación de Householder
        R(j:m, j:n) = R(j:m, j:n) - 2 * vj * (vj' * R(j:m, j:n));

        % Actualiza M aplicando la misma transformación para construir Q
        M(j:m, :) = M(j:m, :) - 2 * vj * (vj' * M(j:m, :));
    end

    % La matriz Q es la transpuesta de M
```

```
    Q = M';
end


% Calcular la descomposición QR
[Q, R] = qr3(A);
```

d)

Cálculo de la factorización implementando los 3 algoritmos

```
% Definición de dimensiones
m = 20; % m = n

% Generar matriz aleatoria
A = rand(m);

% Generar matriz identidad mxm
I = eye(m);

% Algoritmo 1
[Q1,R1] = qr1(A)
```

```
Q1 = 20×20
    0.3307   -0.0977   -0.2383    0.2513   -0.3128    0.1547    0.0172   -0.2536 ···
    0.2868   -0.3824    0.2225    0.1266   -0.0463    0.0835    0.0671   -0.2175
    0.1025   -0.1540    0.3682   -0.0311   -0.0840   -0.1987    0.1880    0.1921
    0.1948    0.0531    0.1931   -0.1867    0.1143    0.2726   -0.2796   -0.2150
    0.3529   -0.3794   -0.1980   -0.3374    0.0693    0.1906   -0.1765    0.3305
    0.2565    0.2049   -0.0818    0.0335    0.2059   -0.3408   -0.2003    0.0420
    0.3007    0.1599    0.0264   -0.0222    0.0195   -0.3347   -0.5950   -0.0216
    0.1553   -0.0641    0.3287    0.2800   -0.1660   -0.0885   -0.0210    0.3202
    0.1687    0.2568    0.0749   -0.0629   -0.1357    0.0033    0.0927   -0.4965
    0.2009    0.2673   -0.2397    0.0592    0.1893    0.1356    0.2194   -0.0961
     ⋮
R1 = 20×20
    2.7903    2.4647    2.5117    2.6984    2.1355    1.7480    1.6743    2.3098 ···
         0    1.4996    0.4459    0.5065    0.9089    0.8411    1.1244    0.5908
         0         0    1.7405    0.6152    0.7914    0.7714    0.4725    1.1057
         0         0         0    1.1607    0.1292    0.0710   -0.2448   -0.1568
         0         0         0         0    1.0818    0.1538    0.6453    0.1562
         0         0         0         0         0    1.0133   -0.0099   -0.4608
         0         0         0         0         0         0    1.0449    0.1244
         0         0         0         0         0         0         0    0.8700
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```

```
% Algoritmo 2
[Q2,R2] = qr1(A)
```

```
Q2 = 20×20
    0.3307   -0.0977   -0.2383    0.2513   -0.3128    0.1547    0.0172   -0.2536 ···
    0.2868   -0.3824    0.2225    0.1266   -0.0463    0.0835    0.0671   -0.2175
```

```
    0.1025   -0.1540    0.3682   -0.0311   -0.0840   -0.1987    0.1880    0.1921
    0.1948    0.0531    0.1931   -0.1867    0.1143    0.2726   -0.2796   -0.2150
    0.3529   -0.3794   -0.1980   -0.3374    0.0693    0.1906   -0.1765    0.3305
    0.2565    0.2049   -0.0818    0.0335    0.2059   -0.3408   -0.2003    0.0420
    0.3007    0.1599    0.0264   -0.0222    0.0195   -0.3347   -0.5950   -0.0216
    0.1553   -0.0641    0.3287    0.2800   -0.1660   -0.0885   -0.0210    0.3202
    0.1687    0.2568    0.0749   -0.0629   -0.1357    0.0033    0.0927   -0.4965
    0.2009    0.2673   -0.2397    0.0592    0.1893    0.1356    0.2194   -0.0961
     ⋮
R2 = 20×20
    2.7903    2.4647    2.5117    2.6984    2.1355    1.7480    1.6743    2.3098 ···
         0    1.4996    0.4459    0.5065    0.9089    0.8411    1.1244    0.5908
         0         0    1.7405    0.6152    0.7914    0.7714    0.4725    1.1057
         0         0         0    1.1607    0.1292    0.0710   -0.2448   -0.1568
         0         0         0         0    1.0818    0.1538    0.6453    0.1562
         0         0         0         0         0    1.0133   -0.0099   -0.4608
         0         0         0         0         0         0    1.0449    0.1244
         0         0         0         0         0         0         0    0.8700
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```

```
% Algoritmo 3
[Q3,R3] = qr1(A)
```

```
Q3 = 20×20
    0.3307   -0.0977   -0.2383    0.2513   -0.3128    0.1547    0.0172   -0.2536 ···
    0.2868   -0.3824    0.2225    0.1266   -0.0463    0.0835    0.0671   -0.2175
    0.1025   -0.1540    0.3682   -0.0311   -0.0840   -0.1987    0.1880    0.1921
    0.1948    0.0531    0.1931   -0.1867    0.1143    0.2726   -0.2796   -0.2150
    0.3529   -0.3794   -0.1980   -0.3374    0.0693    0.1906   -0.1765    0.3305
    0.2565    0.2049   -0.0818    0.0335    0.2059   -0.3408   -0.2003    0.0420
    0.3007    0.1599    0.0264   -0.0222    0.0195   -0.3347   -0.5950   -0.0216
    0.1553   -0.0641    0.3287    0.2800   -0.1660   -0.0885   -0.0210    0.3202
    0.1687    0.2568    0.0749   -0.0629   -0.1357    0.0033    0.0927   -0.4965
    0.2009    0.2673   -0.2397    0.0592    0.1893    0.1356    0.2194   -0.0961
     ⋮
R3 = 20×20
    2.7903    2.4647    2.5117    2.6984    2.1355    1.7480    1.6743    2.3098 ···
         0    1.4996    0.4459    0.5065    0.9089    0.8411    1.1244    0.5908
         0         0    1.7405    0.6152    0.7914    0.7714    0.4725    1.1057
         0         0         0    1.1607    0.1292    0.0710   -0.2448   -0.1568
         0         0         0         0    1.0818    0.1538    0.6453    0.1562
         0         0         0         0         0    1.0133   -0.0099   -0.4608
         0         0         0         0         0         0    1.0449    0.1244
         0         0         0         0         0         0         0    0.8700
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```

Cálculo de las normas con los Q y R de los algoritmos

Para el algoritmo 1:

Cálculo de $\|A - Q_1 R_1\|_2$

```
disp(norm(A - Q1 * R1))
```

```
7.4160e-16
```

Cálculo de $||Q_1Q_1^T - I||_2$

```
disp(norm(Q1 * Q1' - I))
```

```
1.4892e-13
```

Para el algoritmo 2:

Cálculo de $||A - Q_2R_2||_2$

```
disp(norm(A - Q2 * R2))
```

```
7.4160e-16
```

Cálculo de $||Q_2Q_2^T - I||_2$

```
disp(norm(Q2 * Q2' - I))
```

```
1.4892e-13
```

Para el algoritmo 2:

Cálculo de $||A - Q_3R_3||_2$

```
disp(norm(A - Q3 * R3))
```

```
7.4160e-16
```

Cálculo de $||Q_3Q_3^T - I||_2$

```
disp(norm(Q3 * Q3' - I))
```

```
1.4892e-13
```

e)

Cálculo de la factorización implementando los 3 algoritmos con A como la matriz de Hilbert

```
% Definición de dimensiones
m = 20; % m = n

% Generar matriz aleatoria
A = hilb(m);

% Generar matriz identidad mxm
I = eye(m);

% Algoritmo 1
```

```
[Q1,R1] = qr1(A)
```

```
Q1 = 20×20
    0.7915   -0.5565    0.2361   -0.0849    0.0276   -0.0083    0.0023    0.0020 ···
    0.3958    0.2015   -0.6134    0.5399   -0.3249    0.1570   -0.0652   -0.0566
    0.2638    0.2942   -0.3353   -0.1733    0.5125   -0.5153    0.3565    0.3021
    0.1979    0.2926   -0.1232   -0.3587    0.2759    0.1626   -0.4650   -0.3564
    0.1583    0.2725    0.0061   -0.3391   -0.0200    0.3722   -0.2134   -0.2412
    0.1319    0.2499    0.0836   -0.2597   -0.1973    0.2750    0.1484    0.0777
    0.1131    0.2289    0.1303   -0.1718   -0.2697    0.0930    0.3044    0.2717
    0.0989    0.2103    0.1582   -0.0921   -0.2747   -0.0695    0.2768    0.2944
    0.0879    0.1940    0.1747   -0.0247   -0.2416   -0.1798    0.1549    0.2013
    0.0792    0.1799    0.1839    0.0306   -0.1893   -0.2361    0.0106    0.0593
     ⋮
R1 = 20×20
    1.2634    0.7538    0.5568    0.4477    0.3770    0.3269    0.2893    0.2600 ···
         0    0.1737    0.2004    0.1999    0.1923    0.1828    0.1731    0.1640
         0         0    0.0175    0.0298    0.0377    0.0426    0.0455    0.0473
         0         0         0    0.0016    0.0035    0.0053    0.0069    0.0083
         0         0         0         0    0.0001    0.0003    0.0006    0.0009
         0         0         0         0         0    0.0000    0.0000    0.0001
         0         0         0         0         0         0    0.0000    0.0000
         0         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```

```
% Algoritmo 2
[Q2,R2] = qr1(A)
```

```
Q2 = 20×20
    0.7915   -0.5565    0.2361   -0.0849    0.0276   -0.0083    0.0023    0.0020 ···
    0.3958    0.2015   -0.6134    0.5399   -0.3249    0.1570   -0.0652   -0.0566
    0.2638    0.2942   -0.3353   -0.1733    0.5125   -0.5153    0.3565    0.3021
    0.1979    0.2926   -0.1232   -0.3587    0.2759    0.1626   -0.4650   -0.3564
    0.1583    0.2725    0.0061   -0.3391   -0.0200    0.3722   -0.2134   -0.2412
    0.1319    0.2499    0.0836   -0.2597   -0.1973    0.2750    0.1484    0.0777
    0.1131    0.2289    0.1303   -0.1718   -0.2697    0.0930    0.3044    0.2717
    0.0989    0.2103    0.1582   -0.0921   -0.2747   -0.0695    0.2768    0.2944
    0.0879    0.1940    0.1747   -0.0247   -0.2416   -0.1798    0.1549    0.2013
    0.0792    0.1799    0.1839    0.0306   -0.1893   -0.2361    0.0106    0.0593
     ⋮
R2 = 20×20
    1.2634    0.7538    0.5568    0.4477    0.3770    0.3269    0.2893    0.2600 ···
         0    0.1737    0.2004    0.1999    0.1923    0.1828    0.1731    0.1640
         0         0    0.0175    0.0298    0.0377    0.0426    0.0455    0.0473
         0         0         0    0.0016    0.0035    0.0053    0.0069    0.0083
         0         0         0         0    0.0001    0.0003    0.0006    0.0009
         0         0         0         0         0    0.0000    0.0000    0.0001
         0         0         0         0         0         0    0.0000    0.0000
         0         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```

```
% Algoritmo 3
[Q3,R3] = qr1(A)
```

```
Q3 = 20×20
```

```
    0.7915   -0.5565    0.2361   -0.0849    0.0276   -0.0083    0.0023    0.0020 ···
    0.3958    0.2015   -0.6134    0.5399   -0.3249    0.1570   -0.0652   -0.0566
    0.2638    0.2942   -0.3353   -0.1733    0.5125   -0.5153    0.3565    0.3021
    0.1979    0.2926   -0.1232   -0.3587    0.2759    0.1626   -0.4650   -0.3564
    0.1583    0.2725    0.0061   -0.3391   -0.0200    0.3722   -0.2134   -0.2412
    0.1319    0.2499    0.0836   -0.2597   -0.1973    0.2750    0.1484    0.0777
    0.1131    0.2289    0.1303   -0.1718   -0.2697    0.0930    0.3044    0.2717
    0.0989    0.2103    0.1582   -0.0921   -0.2747   -0.0695    0.2768    0.2944
    0.0879    0.1940    0.1747   -0.0247   -0.2416   -0.1798    0.1549    0.2013
    0.0792    0.1799    0.1839    0.0306   -0.1893   -0.2361    0.0106    0.0593
     ⋮
R3 = 20×20
    1.2634    0.7538    0.5568    0.4477    0.3770    0.3269    0.2893    0.2600 ···
         0    0.1737    0.2004    0.1999    0.1923    0.1828    0.1731    0.1640
         0         0    0.0175    0.0298    0.0377    0.0426    0.0455    0.0473
         0         0         0    0.0016    0.0035    0.0053    0.0069    0.0083
         0         0         0         0    0.0001    0.0003    0.0006    0.0009
         0         0         0         0         0    0.0000    0.0000    0.0001
         0         0         0         0         0         0    0.0000    0.0000
         0         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```

Cálculo de las normas con los Q y R de los algoritmos

Para el algoritmo 1:

Cálculo de $||A - Q_1 R_1||_2$

```
disp(norm(A - Q1 * R1))
```

```
    8.4639e-17
```

Cálculo de $||Q_1 Q_1^T - I||_2$

```
disp(norm(Q1 * Q1' - I))
```

```
    12.9716
```

Para el algoritmo 2:

Cálculo de $||A - Q_2 R_2||_2$

```
disp(norm(A - Q2 * R2))
```

```
    8.4639e-17
```

Cálculo de $||Q_2 Q_2^T - I||_2$

```
disp(norm(Q2 * Q2' - I))
```

```
    12.9716
```

Para el algoritmo 2:

Cálculo de $||A - Q_3 R_3||_2$

```
disp(norm(A - Q3 * R3))
```

```
   8.4639e-17
```

Cálculo de $||Q_3 Q_3^T - I||_2$

```
disp(norm(Q3 * Q3' - I))
```

```
   12.9716
```

Factorización con la funcion qr de MATLAB

```
[Q_m, R_m] = qr(A)
```

```
Q_m = 20×20
   -0.7915    0.5565    0.2361   -0.0849   -0.0276    0.0083   -0.0023   -0.0006 ···
   -0.3958   -0.2015   -0.6134    0.5399    0.3249   -0.1570    0.0649    0.0237
   -0.2638   -0.2942   -0.3353   -0.1733   -0.5125    0.5153   -0.3553   -0.1939
   -0.1979   -0.2926   -0.1232   -0.3587   -0.2759   -0.1626    0.4647    0.4877
   -0.1583   -0.2725    0.0061   -0.3391    0.0200   -0.3722    0.2126   -0.1948
   -0.1319   -0.2499    0.0836   -0.2597    0.1973   -0.2750   -0.1490   -0.3661
   -0.1131   -0.2289    0.1303   -0.1718    0.2697   -0.0930   -0.3046   -0.1423
   -0.0989   -0.2103    0.1582   -0.0921    0.2747    0.0695   -0.2766    0.1231
   -0.0879   -0.1940    0.1747   -0.0247    0.2416    0.1798   -0.1545    0.2659
   -0.0792   -0.1799    0.1839    0.0306    0.1893    0.2361   -0.0100    0.2683
     ⋮
R_m = 20×20
   -1.2634   -0.7538   -0.5568   -0.4477   -0.3770   -0.3269   -0.2893   -0.2600 ···
         0   -0.1737   -0.2004   -0.1999   -0.1923   -0.1828   -0.1731   -0.1640
         0         0    0.0175    0.0298    0.0377    0.0426    0.0455    0.0473
         0         0         0    0.0016    0.0035    0.0053    0.0069    0.0083
         0         0         0         0   -0.0001   -0.0003   -0.0006   -0.0009
         0         0         0         0         0   -0.0000   -0.0000   -0.0001
         0         0         0         0         0         0   -0.0000   -0.0000
         0         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
     ⋮
```