```
function [uu, u] = house(x)
% This constructs the unnormalized  vector uu
% defining the Householder reflection that
% zeros all but the first entries in x.
% u is the normalized vector uu/||uu||
%

tol = 2*10^(-15);   % tolerance
uu = x;
p = size(x,1);
% computes l^1-norm of x(2:p,1)
n1 = sum(abs(x(2:p,1)));
if n1 <= tol
   u = zeros(p,1);  uu = u;
else
   l = sqrt(x'*x);  % l^2 norm of x
   uu(1) = x(1) + signe(x(1))*l;
   u = uu/sqrt(uu'*uu);
end
end
```

The Householder transformations are recorded in an array u of $n-1$ vectors. There are more efficient implementations, but for the sake of clarity we present the following version.

```
function [R,  u] = houseqr(A)
%  This function computes the upper triangular R in the QR factorization
%  of A using Householder reflections, and an implicit representation
%  of Q as  a sequence of n - 1 vectors u_i representing Householder
%  reflections

n = size(A, 1);
R = A;
u = zeros(n,n-1);
for i = 1:n-1
    [~, u(i:n,i)] = house(R(i:n,i));
    if u(i:n,i) == zeros(n - i + 1,1)
       R(i+1:n,i) = zeros(n - i,1);
    else
       R(i:n,i:n) = R(i:n,i:n) - 2*u(i:n,i)*(u(i:n,i)'*R(i:n,i:n));
    end
end
end
```