Note: In this problem set, expressions in green cells match corresponding expressions in the text answers.

```
Clear["Global`*"]
```

1 - 5 Householder tridiagonalization
Tridiagonalize.

1. $\begin{pmatrix} 0.98 & 0.04 & 0.44 \\ 0.04 & 0.56 & 0.40 \\ 0.44 & 0.40 & 0.80 \end{pmatrix}$

```
Clear["Global`*"]
```

m1 = $\begin{pmatrix} 0.98 & 0.04 & 0.44 \\ 0.04 & 0.56 & 0.40 \\ 0.44 & 0.40 & 0.80 \end{pmatrix}$

```
{{0.98, 0.04, 0.44}, {0.04, 0.56, 0.4}, {0.44, 0.4, 0.8}}
```

```
A = N[{{0.98`, 0.04`, 0.44`}, {0.04`, 0.56`, 0.4`}, {0.44`, 0.4`, 0.8`}}];
(*A=N[
   {{-42,43,-2,28},{43,-98,72,-26},{-2,72,-96,53},{28,-26,53,54}}];*)
n = Length[A[[1]]];
zeroVector = {};
For[i = 1, i ≤ n, i++, zeroVector = Append[zeroVector, {0}]];
Alist = {A};
Hlist = {};

For[j = 1, j ≤ n - 2, j++, If[A[[j + 1, j]] ≥ 0, c = 1, c = 2];
 alpha = (-1)^c (Sum[A[[k, j]]^2, {k, j + 1, n}])^(1/2);
 r = ((1/2) alpha^2 - (1/2) alpha A[[j + 1, j]])^(1/2);
 x = zeroVector;
 x[[j + 1, 1]] = (A[[j + 1, j]] - alpha) / (2 r);
 For[k = j + 2, k ≤ n, k++, x[[k, 1]] = A[[k, j]] / (2 r)];
 H = IdentityMatrix[n] - 2 x.Transpose[x];
 A = H.A.H;
 Hlist = Append[Hlist, H];
 Alist = Append[Alist, A];]

MatrixForm[Chop[A]]
```

The code seems to work well, and was copied from *https://mathematica.stackexchange.com/questions/46037/mathematica-implementation-of-householder-s-method/115229#115229*, where it was seen in the post of Rikohai. I don't know how to put it into the form of a reusable block or module.

$$\begin{pmatrix} 0.98 & -0.441814 & 0 \\ -0.441814 & 0.870164 & 0.371803 \\ 0 & 0.371803 & 0.489836 \end{pmatrix}$$

3. $\begin{pmatrix} 7 & 2 & 3 \\ 2 & 10 & 6 \\ 3 & 6 & 7 \end{pmatrix}$

$$\text{m2} = \begin{pmatrix} 7 & 2 & 3 \\ 2 & 10 & 6 \\ 3 & 6 & 7 \end{pmatrix}$$

```
{{7, 2, 3}, {2, 10, 6}, {3, 6, 7}}
```

```
A = N[{{7, 2, 3}, {2, 10, 6}, {3, 6, 7}}];
(*A=N[
   {{-42,43,-2,28},{43,-98,72,-26},{-2,72,-96,53},{28,-26,53,54}}];*)
n = Length[A[[1]]];
zeroVector = {};
For[i = 1, i ≤ n, i++, zeroVector = Append[zeroVector, {0}]];
Alist = {A};
Hlist = {};

For[j = 1, j ≤ n - 2, j++, If[A[[j + 1, j]] ≥ 0, c = 1, c = 2];
 alpha = (-1) ^c (Sum[A[[k, j]] ^2, {k, j + 1, n}]) ^ (1 / 2);
 r = ((1 / 2) alpha^2 - (1 / 2) alpha A[[j + 1, j]]) ^ (1 / 2);
 x = zeroVector;
 x[[j + 1, 1]] = (A[[j + 1, j]] - alpha) / (2 r);
 For[k = j + 2, k ≤ n, k++, x[[k, 1]] = A[[k, j]] / (2 r)];
 H = IdentityMatrix[n] - 2 x.Transpose[x];
 A = H.A.H;
 Hlist = Append[Hlist, H];
 Alist = Append[Alist, A];]

MatrixForm[Chop[A]]
```

Again, Rikohai's code duplicates the text's answer.

$$\begin{pmatrix} 7. & -3.60555 & 0 \\ -3.60555 & 13.4615 & 3.69231 \\ 0 & 3.69231 & 3.53846 \end{pmatrix}$$

6 - 9 QR-factorization

Do three QR-steps to find approximations of the eigenvalues of:

7. The matrix in the answer to problem 3.

As the documentation for QRDecomposition states, the result of the operation is a pair of matrices, q an orthogonal matrix to the input matrix, and r an upper diagonal matrix. When combined with the dotting maneuver shown below, a matrix is produced, the diagonal of which consists of approximations of the eigenvalues. Following a lengthy enough iterative series, the approximations become close to actual.

```
Clear["Global`*"]
```

$$m1 = \begin{pmatrix} 7.\text{`} & -3.605551275463989\text{`} & 0 \\ -3.605551275463989\text{`} & 13.46153846153846\text{`} & 3.6923076923076916\text{`} \\ 0 & 3.6923076923076916\text{`} & 3.5384615384615383\text{`} \end{pmatrix}$$

```
{{7., -3.60555, 0}, {-3.60555, 13.4615, 3.69231}, {0, 3.69231, 3.53846}}
```

```
Eigenvalues[m1]
```

```
{16., 6., 2.}
```

```
{q, r} = QRDecomposition[m1]
```

```
{{{-0.889001, 0.457905, 0.}, {-0.431124, -0.837006, -0.336976},
  {-0.154303, -0.299572, 0.941513}}, {{-7.87401, 9.36945, 1.69073},
  {0., -10.9572, -4.28286}, {0., 0., 2.22539}}}
```

```
a = r.Transpose[q]
```

$$\{\{11.2903, -5.01735, 6.66134\times 10^{-16}\},$$
$$\{-5.01735, 10.6144, -0.749906\}, \{0., -0.749906, 2.09524\}\}$$

```
{q1, r1} = QRDecomposition[a]
```

```
{{{-0.913829, 0.4061, 0.}, {-0.404169, -0.909483, 0.0974049},
  {0.0395561, 0.0890114, 0.995245}}, {{-12.355, 8.89552, -0.304536},
  {0., -7.69885, 0.886113}, {0., 0., 2.01852}}}
```

```
a1 = r1.Transpose[q1]
```

$$\{\{14.9028, -3.1265, -2.22045\times 10^{-16}\},$$
$$\{-3.1265, 7.08828, 0.196614\}, \{0., 0.196614, 2.00893\}\}$$

```
{q2, r2} = QRDecomposition[a1]
```

```
{{{-0.978694, 0.205323, 0.}, {-0.205223, -0.978217, -0.0312166},
  {-0.00640949, -0.0305515, 0.999513}}, {{-15.2272, 4.51528, 0.0403694},
  {0., -6.29839, -0.255043}, {0., 0., 2.00194}}}
```

```
a2 = r2.Transpose[q2]
```

$$\{\{15.8299, -1.2932, -6.93889\times 10^{-17}\},$$
$$\{-1.2932, 6.16916, -0.0624937\}, \{0., -0.0624937, 2.00096\}\}$$

The green cells above match the answer in the text for this problem to an accuracy of at least 4S. I was surprised at the closeness of the agreement.

9. $\begin{pmatrix} 140 & 10 & 0 \\ 10 & 70 & 2 \\ 0 & 2 & -30 \end{pmatrix}$

```
Clear["Global`*"]
```

m2 = $\begin{pmatrix} 140 & 10 & 0 \\ 10 & 70 & 2 \\ 0 & 2 & -30 \end{pmatrix}$

```
{{140, 10, 0}, {10, 70, 2}, {0, 2, -30}}
```

```
N[Eigenvalues[m2]]
```

```
{141.401, 68.6392, -30.0402}
```

```
{q, r} = QRDecomposition[m2];
a = r.Transpose[N[q]]
```

$\{\{141.066, 4.92592, 2.77556\times10^{-17}\}, \{4.92592, 68.9666, 0.869129\}, \{0., 0.869129, -30.0326\}\}$

```
{q1, r1} = QRDecomposition[a];
a1 = r1.Transpose[N[q1]]
```

$\{\{141.322, 2.39952, -1.73472\times10^{-17}\}, \{2.39952, 68.717, 0.379731\}, \{0., 0.379731, -30.0388\}\}$

```
{q2, r2} = QRDecomposition[a1];
a2 = N[r2.Transpose[q2]]
```

$\{\{141.382, 1.16574, 5.20417\times10^{-18}\}, \{1.16574, 68.6576, 0.166124\}, \{0., 0.166124, -30.0399\}\}$

The answer in the four green cells above match the text answer to an accuracy of 4S.