1. (**i**)

```
function R = cholesky(A)

% no. of rows of A
n = size(A, 1);

% check symmetry
if ~isequal(A, A')
    error("The input matrix is not symmetric. Cholesky factorization not
        possible.");
else
    fprintf("The input matrix is symmetric. Checking positive-definiteness
        ...\n");
end

% initialize array for storing determinant of leading principal minors
det_arr = zeros(n, 1);

% check positive-definiteness
for i = 1:n
    det_minor = det(A(1:i, 1:i));
    det_arr(i) = det_minor;
    if det_minor <= 0
        error("The input matrix is symmetric, but not positive definite. " +
            ...
            "Cholesky factorization not possible.");
    end
end

if all(det_arr > 0)
    fprintf("The input matrix is SPD.\n");
end

fprintf("Determinants of leading principal minors:\n");
disp(det_arr);

% initialize R
R = triu(A);

% modified Gaussian elimination
for k = 1:n
    for i = k+1:n
        m = R(k, i)/R(k, k);
        R(i, i:n) = R(i, i:n) - m*R(k, i:n);
    end
    R(k, k:n) = R(k, k:n)/sqrt(R(k, k));
end
end
```

Listing 1: Function for *Cholesky* factorization (*cholesky.m*).

(**ii**)

```
function x = forward_backward_sub(R, b)

n = size(R, 1); % no. of rows
R_t = R'; % transpose of R

% initialize y and x vectors
y = zeros(n, 1);
x = zeros(n, 1);

% forward substitution to get y
for i = 1:n
    y(i) = (b(i) - R_t(i, 1:i-1)*y(1:i-1))/R_t(i, i);
end

fprintf("y = \n");
disp(y);

% backward substitution to get x
for i = n:-1:1
    x(i) = (y(i) - R(i,i+1:n)*x(i+1:n)) / R(i,i);
end
end
```

Listing 2: Function for forward and backward substitution to find $\vec{y}$ and $\vec{x}$ respectively (*forward_backward_sub.m*).

(**iii**)

```
clear all; clc;

% inputs
A = [4 1 1 1; 1 3 -1 1; 1 -1 2 0; 1 1 0 2];
b = [0.6; 0; 0; 0.4];

% call "cholesky()" to get R
R = cholesky(A);
fprintf("R = \n");
disp(R);
fprintf("R* = \n");
disp(R');
% verify A = R*R
fprintf("A = \n");
disp(R'*R);

% call "forward_backward_sub()" to get x
x = forward_backward_sub(R, b);

fprintf("x = \n");
disp(x);
```

Listing 3: Input-output script to solve for *Cholesky* factorization (*input_output.m*).

```
The input matrix is symmetric. Checking positive-definiteness...
The input matrix is SPD.
Determinants of leading principal minors:
     4
    11
    13
    20

R =
    2.0000   0.5000   0.5000   0.5000
         0   1.6583  -0.7538   0.4523
         0        0   1.0871   0.0836
         0        0        0   1.2403

R* =
    2.0000        0        0        0
    0.5000   1.6583        0        0
    0.5000  -0.7538   1.0871        0
    0.5000   0.4523   0.0836   1.2403

A =
    4.0000   1.0000   1.0000   1.0000
    1.0000   3.0000  -1.0000   1.0000
    1.0000  -1.0000   2.0000   0.0000
    1.0000   1.0000   0.0000   2.0000

y =
    0.3000
   -0.0905
   -0.2007
    0.2481

x =
    0.2000
   -0.2000
   -0.2000
    0.2000
```

Listing 4: Output terminal.

So, the solution of the linear system:

$$\vec{x} = \begin{pmatrix} 0.2 \\ -0.2 \\ -0.2 \\ 0.2 \end{pmatrix}$$