Ex 1;

```
A=[1 2 3; 2 5 8; 3 8 14];
b=[(-5);(-14);(-25)];
[L1]=FactCholesky(A,b)
[L2]=FactCholesky2(A,b)
L3=chol(A,'lower')%Verificare L
L1*transpose(L1);%Verificare L
X=inv(A)*b%Verificare rezultatele ecuatiei Ax=b
type('FactCholesky')
type('FactCholesky2')
type('SubsDesc')
type('SubsAsc')
```

```
y =

    -5
    -4
    -2


x =

     1
     0
    -2


L1 =

     1     0     0
     2     1     0
     3     2     1


y =

    -5
    -4
    -2


x =

     1
     0
    -2


L2 =
     1     0     0
     2     1     0
     3     2     1


L3 =

     1     0     0
     2     1     0
     3     2     1


X =

    1.0000
   -0.0000
   -2.0000


function [L] = FactCholesky(A,b)
%Factorizarea Cholesky metoda 1
a=A(1,1);
[m,n]=size(A);
if(a<=0)
    error('A nu este pozitiv definita')
    return
end
L(1,1)=sqrt(A(1,1));
for(i=2:n)
    L(i,1)=A(i,1)/L(1,1);
end
for(k=2:n)
    suma=0;
    for(s=1:k-1)
        suma=suma+L(k,s)^2;
    end
    a=A(k,k)-suma;
    if(a<=0)
        error('A nu este pozitiv definita')
        return
    end
    L(k,k)=sqrt(a);
    for(i=k+1:n)
        suma=0;
        for(s=1:k-1)
            suma=suma+L(i,s)*L(k,s);
        end
        L(i,k)=1/L(k,k)*(A(i,k)-suma);
    end
end
y=SubsAsc(L,b)
Lt=transpose(L);
x=SubsDesc(Lt,y)
end
```

```
function [L] = FactCholesky2(A,b)
%Factorizarea Cholesky metoda 2
a=A(1,1);
[m,n]=size(A);
if(a<=0)
    error('A nu este pozitiv definita')
    return
end
for i=2:n
    L(i,1)=A(i,1)/sqrt(A(1,1));
end
for k=1:n
    suma=0;
    for j=1:k-1
        suma=suma+L(k,j)^2;
    end
    L(k,k)=sqrt(A(k,k)-suma);
    for i=k+1:n
        suma=0;
        for j=1:k-1
            suma=suma+L(i,j)*L(k,j);
        end
        L(i,k)=(A(i,k)-suma)/L(k,k);
    end
end
y=SubsAsc(L,b)
Lt=transpose(L);
x=SubsDesc(Lt,y)
end

function[x]=SubsDesc(A,b)
%SubsDesc rezolva sisteme superior triunghiulare

%Synopsis: [x]=SubsDesc(A,b)
%Input: A=matrice superior triunghiulara (Aij=0,i>j)
%       b=vectorul termenilor liberi
%Output:x=vectorul solutie
[m,n]=size(A);
%se verifica daca matricea 'A' este patratica
if m~=n
    error('A nu este patratica')
    return
end
%se verifica daca matricea 'A' este superior triunghiulara
for i=2:m
    for j=1:i-1
        if A(i,j)~=0
            error('A nu este superior triunghiulara')
            return
        end
    end
end
```

```
%se verifica daca sistemul este compatibil determinat
for i=1:m
    if A(i,i)==0
        error('Sistemul nu admite solutie unica')
        return
    end
end
%se verifica compatibilitate dimensiuni
if length(b)~=n
    error('Incompatibilitate intre dimensiuni')
    return
end
x(n)= b(n)/A(n,n);
for  k=n-1:-1:1
    sum=0;
    for j=k+1:n
        sum=sum+A(k,j)*x(j);
    end
    x(k)=1/A(k,k)*(b(k)-sum);
end
x=transpose(x);

function[x]=SubsAsc(A,b)
[m,n]=size(A);
x(1)=1/A(1,1)*b(1);
for(k=2:n)
    suma=0;
    for(j=1:k-1)
        suma=suma+A(k,j)*x(j);
    end
    x(k)=1/A(k,k)*(b(k)-suma);
end
x=transpose(x);
end
```

Ex 2;

```
n=5;
for(i=1:n)
    a(i)=2*n-2*(i-1);
    b(i)=i^2;
end
for(i=1:n)
    for(j=1:n)
        A(i,i)=a(1);
        if(j>i)
            A(i,j)=a(j-i+1);
        end
        if(j<i)
            A(i,j)=a(i-j+1);
        end
    end
```

```
end
a=transpose(a);

%Criteriul Sylvester
for(i=1:n)
if(det(A([1:i],[1:i]))<0)
    error('Matricea nu este pozitiv definita')
    return;
end
end
fprintf('Matricea este pozitiv definita')
[L1]=FactCholesky(A,b)
L2=chol(A,'lower')%Verificare L
L1*transpose(L1);%Verificare L
X=inv(A)*b%Verificare rezultatele ecuatiei Ax=b

Matricea este pozitiv definita
y =

    0.3162
    1.6865
    2.9463
    4.4098
    6.1721


x =

    0.3333
   -0.5000
   -0.5000
   -0.5000
    3.3333


L1 =

    3.1623         0         0         0         0
    2.5298    1.8974         0         0         0
    1.8974    1.6865    1.8856         0         0
    1.2649    1.4757    1.6499    1.8708         0
    0.6325    1.2649    1.4142    1.6036    1.8516


L2 =

    3.1623         0         0         0         0
    2.5298    1.8974         0         0         0
    1.8974    1.6865    1.8856         0         0
    1.2649    1.4757    1.6499    1.8708         0
    0.6325    1.2649    1.4142    1.6036    1.8516


X =
    0.3333
   -0.5000
   -0.5000
   -0.5000
    3.3333
```

Published with MATLAB® R2019a