Ex 1.

```
A1=[0 1 1; 2 1 5; 4 2 1];
b1=[3;5;1];
A2=[0 1 (-2); 1 (-1) 1; 1 0 (-1)];
b2=[4;6;2];

type('GaussPivTotala')

[x1]=GaussPivTotala(A1,b1)
if A1*x1==b1
    fprintf('Solutie a fost calculata corect\n')
else
    fprintf('Solutie gresita')
end
%Pentru ca pentru al doilea exemplu ar da eroare am ales sa nu il
 rulez si
%doar sa il comentez.

% [x2]=GaussPivTotala(A2,b2);
% if A2*x2==b2
%     fprintf('Solutie a fost calculata corect\n')
% else
%     fprintf('Solutie gresita')
% end
```

```
function [x,c,val] = GaussPivTotala(A,b)
%GaussPivTotala rezolva sisteme patratice de ecuatii liniare

%Synopsis: x=GaussPivTotala(A,b)
%Input: A=matrice patratica
%       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 compatibilitate dimensiuni
if length(b)~=n
    error('Incompatibilitate intre dimensiuni')
    return
end
[m,n]=size(A);
a(1)=1;
for sol=2:n
    a(sol)=a(sol-1)+1;
end
Ae=A;
for i=1:n
```

```
    Ae(i,n+1)=b(i);
end
contuor=0;
for piv=1:n
    maxpivot=Ae(piv,piv);
    p=piv;
    m=piv;
    for j=piv:n
    for i=piv:n
        if maxpivot<abs(Ae(i,j))
            maxpivot=abs(Ae(i,j));
            p=i;
            m=j;
        end
    end
    end
         if maxpivot==0
            error('Sistemul este incompatibil sau compatibil
 nedeterminat')
            return
         end
         if p~=piv
            Ae([p,piv],:)=Ae([piv,p],:);
            contuor=contuor+1;
         end
         if m~=piv
            Ae(:,[m,piv])=Ae(:,[piv,m]);
            a([m piv])=a([piv m]);%memorez intr-un vector cum se
 schimba coloanele, implicit valorile
            contuor=contuor+1;
         end
         for l=piv+1:n
             mlk=Ae(l,piv)/Ae(piv,piv);
             Ae(l,:)=Ae(l,:)-mlk*Ae(piv,:);
         end
end
if Ae(n,n)==0
    error('Sistemul este incompatibil sau compatibil nedeterminat')
    return
end
A=Ae(1:n,1:n);
b=Ae(:,n+1);
x=SubsDesc(A,b);
%Aici am facut rearanjarea solutiilor in cazul in care s-au
 interschimbat
%coloanele.
for i=1:n
    val(i)=Ae(i,i);
end
c=contuor;
for sol=1:n
    if a(sol)~=sol
        x([sol a(sol)])=x([a(sol) sol]);
        a([sol a(sol)])=a([a(sol) sol]);
    end
end
end

x1 =

    -1
     2
     1

Solutie a fost calculata corect
```

Ex 2.

```
A=0;
b=0;
n=10;
d=12;f=(-4);c=(-4);
for  i=1:n
    for j=1:n
        if i==j
            A(i,j)=d;
        elseif j==i-1
            A(i,j)=c;
        elseif j==i+1
            A(i,j)=f;
        else
            A(i,j)=0;
        end
    end
end
for i=1:n
    if (i==1 || i==n)
    b(i,1)=2;
    else
        b(i,1)=1;
    end
end

type('GaussPivTotala')

[x1]=GaussPivTotala(A,b)
if A*x1==b
    fprintf('Solutie a fost calculata corect\n')
else
    fprintf('Solutie gresita')
end
% Observatie:
% Nu am putut sa imi dau seama de ce pentru n=5,7,8 si 10
% programul imi da "Solutie gresita" chiar daca rezultatele sunt tot
 0.250
% iar n-ul nu ar trebui sa influenteze solutiile. Deci programul ar
 trebui
```

```
% sa functioneze pentru orice n din moment ce pentru n de la 1-11 nu
% functioneaza cu 5,7,8,10. Puteti incerca sa vedeti, si daca se poate
 sa
% explicati ce am putut gresi.


%Ex 3.

type('InversDet')

[invA,detA]=InversDet(A)
x2=invA*b

%Continuare a observatie, solutiile ar trebui sa fie corecte zinand
 cont ca
%am rezolvat sistemul folosind inversa matrice si mi-a dat acelasi set
 de
%valor.
```

```
function [x,c,val] = GaussPivTotala(A,b)
%GaussPivTotala rezolva sisteme patratice de ecuatii liniare

%Synopsis: x=GaussPivTotala(A,b)
%Input: A=matrice patratica
%       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 compatibilitate dimensiuni
if length(b)~=n
    error('Incompatibilitate intre dimensiuni')
    return
end
[m,n]=size(A);
a(1)=1;
for sol=2:n
    a(sol)=a(sol-1)+1;
end
Ae=A;
for i=1:n
    Ae(i,n+1)=b(i);
end
contuor=0;
for piv=1:n
    maxpivot=Ae(piv,piv);
    p=piv;
    m=piv;
    for j=piv:n
    for i=piv:n
```

```
        if maxpivot<abs(Ae(i,j))
            maxpivot=abs(Ae(i,j));
            p=i;
            m=j;
        end
    end
    end
         if maxpivot==0
            error('Sistemul este incompatibil sau compatibil
 nedeterminat')
            return
         end
         if p~=piv
            Ae([p,piv],:)=Ae([piv,p],:);
            contuor=contuor+1;
         end
         if m~=piv
            Ae(:,[m,piv])=Ae(:,[piv,m]);
            a([m piv])=a([piv m]);%memorez intr-un vector cum se
 schimba coloanele, implicit valorile
            contuor=contuor+1;
         end
         for l=piv+1:n
             mlk=Ae(l,piv)/Ae(piv,piv);
             Ae(l,:)=Ae(l,:)-mlk*Ae(piv,:);
         end
end
if Ae(n,n)==0
    error('Sistemul este incompatibil sau compatibil nedeterminat')
    return
end
A=Ae(1:n,1:n);
b=Ae(:,n+1);
x=SubsDesc(A,b);
%Aici am facut rearanjarea solutiilor in cazul in care s-au
 interschimbat
%coloanele.
for i=1:n
    val(i)=Ae(i,i);
end
c=contuor;
for sol=1:n
    if a(sol)~=sol
        x([sol a(sol)])=x([a(sol) sol]);
        a([sol a(sol)])=a([a(sol) sol]);
    end
end
end

x1 =

    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
```

Solutie gresita

```
function [invA,detA]=InversDet(A);
[m,n]=size(A);
contor=0;
detA=1;
o=0;
for i=1:m
    for j=1:n
        if i==j
            I(i,i)=1;
        else I(i,j)=0;
        end
    end
end
for j=1:n
    [invA(:,j),c,val]=GaussPivTotala(A,I(:,j));
    contor=contor+c;
    if(o==0)
        for i=1:n
            detA=detA*val(i);
        end
o=1;
    end
end
S=contor; detA=((-1)^contor)*detA;
end
```

```
invA =

  Columns 1 through 7

    0.0955    0.0365    0.0139    0.0053    0.0020    0.0008    0.0003
    0.0020    0.0061    0.0163    0.0427    0.1118    0.0427    0.0163
    0.0139    0.0418    0.1115    0.0426    0.0163    0.0062    0.0024
    0.0053    0.0160    0.0426    0.1118    0.0427    0.0163    0.0062
    0.0365    0.1094    0.0418    0.0160    0.0061    0.0023    0.0009
    0.0008    0.0023    0.0062    0.0163    0.0427    0.1118    0.0427
    0.0003    0.0009    0.0024    0.0062    0.0163    0.0427    0.1118
    0.0001    0.0003    0.0009    0.0024    0.0062    0.0163    0.0426
    0.0000    0.0001    0.0003    0.0009    0.0023    0.0061    0.0160
    0.0000    0.0000    0.0001    0.0003    0.0008    0.0020    0.0053

  Columns 8 through 10

    0.0001    0.0000    0.0000
    0.0062    0.0023    0.0008
    0.0009    0.0003    0.0001
    0.0024    0.0009    0.0003
    0.0003    0.0001    0.0000
    0.0163    0.0061    0.0020
    0.0426    0.0160    0.0053
    0.1115    0.0418    0.0139
    0.0418    0.1094    0.0365
    0.0139    0.0365    0.0955


detA =

   1.8571e+10


x2 =

    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
    0.2500
```

Published with MATLAB® R2019a