```
% Esercitazione di laboratorio del 02/11/2020
format short e;format compact
close all;clear all
maxit=10000;
tol=1e-8;

% Es 1,2,3   (il grafico di x non e' riportato)

disp([ '    n1                    Jacobi                   GaussSeidel'])
disp([ '                 its           Time           its         Time'])
for n1=10:10:60,
     [A,f]=poisson2d(n1);
     n=size(A,1);
     x0=zeros(n,1);
     tic;
     [xJ,itsJ,normrJ] = jacobi(A,f,x0,maxit,tol);
     timeJ=toc;
     figure(n1)
     semilogy(1:itsJ,normrJ/normrJ(1),'r')
     hold on
     tic;
     [xGS,itsGS,normrGS] = GaussSeidel(A,f,x0,maxit,tol);
     timeGS=toc;
     semilogy(1:itsGS,normrGS/normrGS(1),'k')
     hold off
     title(['dim:' num2str(n)])
     xlabel('numero di iterazioni')
     ylabel('norma relativa del residuo')
     legend('Jacobi','Gauss-Seidel')
disp([ n1, itsJ, timeJ, itsGS, timeGS])
%pause
end

% Es 4
     n1=10;
     [A,f]=poisson2d(n1);
     n=size(A,1);
     x0=zeros(n,1);
     f=ones(n,1);
     B_J=diag(diag(A))\(tril(A,-1)+triu(A,1));
     rhoJ=max(abs(eig(full(B_J))));
     B_GS=tril(A)\triu(A,1);
     rhoGS=max(abs(eig(full(B_GS))));
     figure(15)
     [xJ,itsJ,normrJ] = jacobi(A,f,x0,maxit,tol);
     semilogy(1:itsJ,normrJ/normrJ(1),'r')
     hold on
     semilogy(rhoJ.^(0:500),'r--')
     [xGS,itsGS,normrGS] = GaussSeidel(A,f,x0,maxit,tol);
     semilogy(1:itsGS,normrGS/normrGS(1),'k')
     semilogy(rhoGS.^(0:500),'k--')
     hold off
     legend('Jacobi','rho J','Gauss-Seidel','rho GS')
     fprintf('\n')
     fprintf('rho J:  %d    rho GS:  %d  \n',rhoJ,rhoGS)


%%%%%%%%%%%%%%%%%%% CUT HERE %%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [x,its,normr] = jacobi(A,b,x0,maxit,tol)

r = b-A*x0;
x = x0;
normr0=norm(r);
normr(1)=normr0;
```

```
p=diag(A);
its=1;

%disp(['num.iter   residuo'])
while normr(its)/normr0 > tol

    x = x + r./p;
    r = b - A*x;
    its = its+1;
    normr(its,1)=norm(r);
%disp([its,normr(its)])
    if its>=maxit, break,end

end



%%%%%%%%%%%%%%%%%% CUT HERE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%



function [x,its,normr] = GaussSeidel(A,b,x0,maxit,tol)

r = b-A*x0;
x = x0;
normr0=norm(r);
normr(1)=normr0;
P=tril(A);
its=1;

%disp(['num.iter   residuo'])
while normr(its)/normr0 > tol

    x = x + P\r;    % Usare propria funzione di eliminaz.Gauss
    r = b - A*x;
    its = its+1;
    normr(its,1)=norm(r);
%disp([its,normr(its)])
    if its>=maxit, break,end

end
```