```
format short e;format compact
clear all;close all;
n=10;
autovalori=1:n;
tipo=input('A sim o nonsim (1/2)\n');

switch tipo

   case 2
   % es 1
      Q=randn(n,n);
      A=Q*diag(autovalori)/Q;

   case 1
   % es 2
      Q=orth(randn(n,n));
      A=Q*diag(autovalori)*Q';
end

maxit=500;
tol=1e-8;
x0=rand(n,1);x0=x0/norm(x0);

fprintf('Approx autovalore piu grande in modulo\n')
%[x,lambda]=potenze(A,x0,maxit,tol);
figure(1)
[x,lambda]=potenze(A,x0,maxit,tol,autovalori(n));

semilogy( (autovalori(n-1)/autovalori(n)).^(1:180),'kx')
legend('residuo','errore','fattore asintot')

hold off

%es 3.i
fprintf('Approx autovalore piu piccolo in modulo\n')
figure(2)
[x,lambda,~]=potenze_inv(A,0,x0,maxit,tol,autovalori(1));
semilogy( (autovalori(1)/autovalori(2)).^(1:40),'kx')
legend('residuo','errore','fattore asintot')

%es 3.ii
fprintf('Approx secondo autovalore piu piccolo in modulo\n')
figure(2)
[x,lambda,tot_iter]=potenze_inv(A,0,x0,maxit,tol,autovalori(2));
semilogy( (autovalori(1)/autovalori(2)).^(1:40),'kx')
legend('residuo','errore','fattore asintot')

fprintf('Approx secondo autovalore piu piccolo in modulo\n')
figure(3)
store_iter=[];
for mu=1.55:0.05:1.95
   [x,lambda,tot_iter]=potenze_inv(A,mu,x0,maxit,tol,autovalori(2));
   store_iter=[store_iter,tot_iter];
   semilogy( abs( (autovalori(2)-mu)./(autovalori(1)-mu)).^(1:40),'kx')
   legend('residuo','errore','fattore asintot')
end

figure(4)
mu=1.55:0.05:1.95;
plot(abs( (autovalori(2)-mu)./(autovalori(1)-mu)),store_iter,'-*k')
xlabel('|\lambda_2 - \mu|/|\lambda_1-\mu|')
title('num di iterazioni rispetto al rapporto asintotico')

figure(5)
mu=1.55:0.05:1.95;
plot(mu,store_iter,'-*')
```

```
xlabel('\mu')
title('num di iterazioni rispetto a \mu ')

return


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

function [x,lambda]=potenze(A,x0,maxit,tol,etrue)

x=x0/norm(x0);
y=A*x;
store_res=[];
store_eig=[];

for i=0:maxit
        lambda = x'*y;
        r=y-lambda*x;
        relres=norm(r)/abs(lambda);
        store_res=[store_res,relres];
        store_eig=[store_eig,abs(lambda-etrue)];
        disp([i,relres,abs(lambda)])
        if relres < tol
                break
        end
        x=y/norm(y);
        y=A*x;
end

semilogy(store_res)
hold on
semilogy(store_eig,'r')
legend('residuo','errore')

end


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

function [x,truelambda,tot_iter]=potenze_inv(A,mu,x0,maxit,tol,etrue)

x=x0/norm(x0);
I=speye(size(A));
[L,U]=lu(A-mu*I);
%y=A*x;
y=U\(L\x);
store_res=[];
store_eig=[];

for i=0:maxit
        lambda = x'*y;
        r=y-lambda*x;
        relres=norm(r)/abs(lambda);
        store_res=[store_res,relres];
        truelambda=1/lambda+mu;
        store_eig=[store_eig,abs(truelambda-etrue)];
        disp([i,relres,abs(truelambda)])
        if relres < tol
                tot_iter=i;
                break
        end
        x=y/norm(y);
        %y=A*x;
        y=U\(L\x);
end
```

```
semilogy(store_res)
hold on
semilogy(store_eig,'r')
legend('residuo','errore')

end
```