Problema 1 *Să se rezolve sistemul $n \times n$*

$$\begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & & \vdots \\ 0 & -1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2 & -1 & 0 \\ \vdots & & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

cu metodele Jacobi, Gauss-Seidel şi SOR, pentru $n = 100$, cu o precizie de 10^{-6}.

Problema 2 *Iteraţia inversă este un algoritm pentru calculul celei mai mici valori proprii (în modul) a unei matrice simetrice A:*

Choose x_0
for** $k = 1, 2, \ldots, m$ (until convergence) **do
solve $Ax_{k+1} = x_k$
normalize $x_{k+1} := x_{k+1}/\|x_{k+1}\|$
end for

Atunci $\lambda = x_m^T A x_m / x_m^T x_m$ este o aproximare a celei mai mici valori proprii. O implementare simplă a acestui algoritm este

```
x=rand(n,1)
for k= 1:m
  x=A\x;
  x=x/norm(x);
  if convergence, break; end;
end
lambda=x'*A*x
```

Pentru matrice mari, putem face economie de operaţii dacă calculăm descompunerea LU a matricei A o singura dată. Iteraţia se realizează utilizând factorii L şi U. În acest mod, fiecare iteraţie necesită doar $O(n^2)$ operaţii, în loc de $O(n^3)$ în programul de mai sus. Utilizaţi funcţiile dumneavoastra pentru descompunere LUP, substituţie directă şi inversă pentru a implementa iteraţia inversă. Experimentaţi cu câteva matrice şi comparaţi rezultatele dumneavoastră cu cele furnizate de `eig(A)`.