

```

1 #ifndef TOOL_H
2 #define TOOL_H
3
4 #include <stdlib.h>
5 #include <stdio.h>
6 #include <math.h>
7
8 // Tolleranza per controlli sui numeri decimali
9 #define TOL 1e-10
10
11 // Typedef per brevità
12 typedef long double Elemento;
13 typedef Elemento *Vettore;
14 typedef Vettore *Matrice;
15
16 Matrice allocaMatrice(const unsigned ndim);           // Alloca una matrice quadrata di
// tipo Elemento di dimensione ndim x ndim
17 void liberaMatrice(Matrice m, const unsigned ndim); // Libera la matrice allocata
// tramite allocaMatrice
18 Vettore allocaVettore(const unsigned ndim);           // Alloca un vettore di Elemento
// di dimensione ndim
19
20 // Esegue la fattorizzazione LU in-place. Alloca e restituisce un vettore per
// ricostruire il pivoting
21 int *fattorizzazioneLU(Matrice *m, const unsigned ndim);
22
23 // Esegue la risoluzione di un sistema lineare con matrice triangolare superiore.
// Alloca e restituisce la soluzione
24 Vettore backwardSubstitution(const Matrice a, const Vettore y, const unsigned ndim);
25
26 // Esegue la risoluzione di un sistema lineare con matrice triangolare superiore.
// Alloca e restituisce la soluzione
27 Vettore forwardSubstitution(const Matrice a, const Vettore y, const unsigned ndim);
28
29 // Esegue il pivoting di un vettore, Alloca e restituisce il vettore permutato
30 Vettore permVett(const int *const pivot, const Vettore y, const unsigned ndim);
31
32 // Risolve il sistema lineare dato dalla matrice quadrata A di dimensione ndim x ndim
// e dal vettore dei termini noti y
33 Vettore risolviSistemaLineare(const Matrice A, const Vettore y, const unsigned ndim);
34
35 #endif

```