

```

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
17 // di tipo Elemento di dimensione ndim x ndim
18 void liberaMatrice(Matrice m, const unsigned ndim);  // Libera la matrice allocata
19 // tramite allocaMatrice
20 Vettore allocaVettore(const unsigned ndim);           // Alloca un vettore di
21 // Elemento di dimensione ndim
22
23 // Esegue la fattorizzazione LU in-place. Alloca e restituisce un vettore per
24 // ricostruire il pivoting
25 int* fattorizzazioneLU(Matrice* m, const unsigned ndim);
26
27 // Esegue la risoluzione di un sistema lineare con matrice triangolare superiore.
28 // Alloca e restituisce la soluzione
29 Vettore backwardSubstitution(const Matrice a, const Vettore y, const unsigned ndim);
30
31 // Esegue la risoluzione di un sistema lineare con matrice triangolare superiore.
32 // Alloca e restituisce la soluzione
33 Vettore forwardSubstitution(const Matrice a, const Vettore y, const unsigned ndim);
34
35 // Esegue il pivoting di un vettore, Alloca e restituisce il vettore permutato
36 Vettore permVett(const int* const pivot, const Vettore y, const unsigned ndim);
37
38 // Risolve il sistema lineare dato dalla matrice quadrata A di dimensione ndim x ndim
39 // e dal vettore dei termini noti y
40 Vettore risolviSistemaLineare(const Matrice A, const Vettore y, const unsigned ndim);
41
42 #endif

```