

Esercizio lode

(1) Esercizio lode

ESSAY marked out of 1 penalty 0 File picker

Data una matrice quadrata A di dimensione N , in molteplici contesti (e.g., studio di sistemi di equazioni lineari, calcolo infinitesimale multi-dimensionale, calcolo tensoriale, calcolo combinatorio) si rende necessario calcolare il determinante della matrice $\det(A)$. Esistono vari modi per calcolare il determinante. Un modo semplice consiste nel trasformare la matrice in *triangolare superiore* e una volta ottenuta la matrice triangolare superiore, il determinante è il prodotto degli elementi della diagonale $\det(A) = \prod_{i=0, N-1} ts(A)_{i,i}$ dove $ts(A)$ è la matrice triangolare superiore di A . Una matrice A è in formato *triangolare superiore* se $A_{i,j} = 0$ per ogni $i > j$.

Uno dei metodi noti (più semplici) per trasformare una matrice A in triangolare superiore consiste nell'applicare il metodo noto con il nome di *Gauss Elimination*.

Il metodo consiste nel modificare ogni riga k della matrice A come segue:

- per ogni riga i tale che i è maggiore di k e per ogni j maggiore o uguale a k , l'elemento $A_{i,j}$, sotto l'assunzione che $A_{k,k} \neq 0$, $A_{i,j} = A_{i,j} - (A_{i,k}/A_{k,k}) * A_{k,j}$.

Si chiede di implementare una procedura `determinante` che prende come argomento una matrice quadrata A di `float`, ed un intero N che rappresenta la dimensione della matrice, la triangolarizza e poi ne calcola il determinante.

La procedura `determinante` può operare in modo **distruttivo** sulla matrice A trasformandola in triangolare superiore sfruttando il metodo della Gauss Elimination discusso qui sopra. Se invece si usano matrici di supporto (da allocare dinamicamente) devono poi essere liberate prima di ritornare il determinante per evitare memory leak.

La funzione `determinante` è inserita in un semplice programma che alloca una matrice $N * N$ con valori random, stampa la matrice generata, chiama la funzione `determinante` per calcolarne il determinante, e la matrice poi viene deallocata. Il `main` e le altre funzioni già presenti nel file `lode.cpp` **NON DEVONO ESSERE MODIFICATE**. Un esempio di esecuzione è il seguente:

```
marco > ./a.out
Matrice A[6][6]
    12      8      17      18      4      14
     8     14     12     19     11     13
    18     13     16      2      5     15
     6     17     13     12     19      5
     7      6     17      8      2      2
    16      6     11     13     16      7
det(A) = 1.49e+06
```

Note:

- A causa di imprecisione della rappresentazione `float` nell'output potrebbero presentarsi degli elementi $A[i][j]$ con $i \leq j$ che non sono esattamente 0.0, ma un valore prossimo allo zero!
- Scaricare il file `lode.cpp`, modificarlo per inserire la definizione della funzione `triangolarizza`, e caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio nello spazio apposito.

- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `iomanip`, `cstdlib` e `ctime`.
- Si ricorda che, l'esempio di esecuzione è puramente indicativo, e la soluzione proposta **NON** deve funzionare solo per l'esempio proposto, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma.

lode.cpp

Information for graders:

Total of marks: 1