

Laboratorio di Programmazione e Calcolo
Canale 2
Appunti del corso

Simone Cacace e Giuseppe Visconti

Dipartimento di Matematica
Sapienza Università di Roma

Anno Accademico 2025–2026

Laboratorio – Array

Esercizi

E1. Somma incrociata

Inizializzati due vettori a e b di interi della stessa lunghezza n , calcolare la somma incrociata degli elementi: $a[1]+b[n]$, $a[2]+b[n-1]$, ... e memorizzarla nel vettore c . Visualizzare i vettori.

Suggerimento: allocare dinamicamente i vettori a , b e c con `malloc`; caricare i valori con un ciclo `for` e calcolare ciascun elemento di $c[i]$ come somma incrociata; ricordarsi di liberare la memoria con `free`.

E2. Statistiche su voti di studenti

Memorizza in un array di interi i voti ottenuti da n studenti e determina il maggiore, il minore e la media.

Suggerimento: creare un array dinamico di interi e leggere i voti con un ciclo `for`; usare variabili per mantenere il massimo, il minimo e la somma; calcolare la media come valore reale (`double`) dividendo per n .

E3. Classifica concorrenti su due prove

Carica i punteggi di n concorrenti su due prove (punteggio minimo 1, punteggio massimo 10) e determina la classifica.

Suggerimento: usare una matrice dinamica che contiene sulla prima colonna l'ID del concorrente e sulla seconda colonna la somma delle due prove; ordinare le righe della matrice in ordine decrescente rispetto alla seconda colonna.

E4. Manipolazione di vettori

Scrivere un programma che:

- letti gli elementi di un vettore $v1$ e un numero k , determini l'elemento di $v1$ più prossimo a k ;
- letto un secondo vettore $v2$ di lunghezza uguale a $v1$, determini il vettore $w1$ di lunghezza doppia rispetto ai precedenti ottenuto alternando gli elementi di $v1$ e $v2$;
- inizialli un vettore $w2$ di lunghezza uguale a $v1$ con i valori

$$\begin{cases} w_2(i) = 1 & \text{se } v_1(i) > v_2(i) \\ w_2(i) = 0 & \text{se } v_1(i) = v_2(i) \\ w_2(i) = -1 & \text{se } v_1(i) < v_2(i) \end{cases}$$

Suggerimento: allocare dinamicamente tutti i vettori; usare un ciclo per confrontare gli elementi di $v1$ e $v2$; per creare $w1$, inserire alternativamente gli elementi dei due vettori; ricordarsi di liberare tutta la memoria a fine programma.

E5. Manipolazione di matrici

Scrivere un programma che:

- inizializzi e quindi visualizzi una matrice di interi in cui ciascun elemento è dato dalla somma dei propri indici;
- letta una matrice quadrata di interi, controlli se è simmetrica;
- inizializzata una matrice quadrata, visualizzi la matrice trasposta;
- letta una matrice di interi o reali positivi, individui una colonna in cui la somma degli elementi ha il valore maggiore;

Suggerimento: rappresentare la matrice come array dinamico; allocare ogni riga con `malloc`; per la simmetria, confrontare `a[i][j]` e `a[j][i]`; per la trasposta, creare una nuova matrice e scambiare righe e colonne; per la colonna massima, usare doppi cicli `for`.

E6. Prodotto di array

Scrivere un programma che:

- calcoli il prodotto scalare tra due vettori di dimensioni consistenti;
- calcoli il prodotto matrice per vettore con dimensioni consistenti;
- calcoli il prodotto di due matrici con dimensioni consistenti.

Suggerimento: verificare la compatibilità tra le dimensioni prima del calcolo; usare array dinamici per tutti i vettori e le matrici; per i prodotti, usare cicli annidati e un accumulatore per la somma parziale; liberare sempre la memoria al termine.

E7. Crivello di Eratostene

Il Crivello di Eratostene è un antico algoritmo (attribuito al matematico greco Eratostene di Cirene, III secolo a.C.) per trovare tutti i numeri primi fino a un certo limite n . L'idea è la seguente:

- Si scrivono tutti i numeri interi da 2 a n .
- Si parte dal primo numero non eliminato (che è primo), e si eliminano tutti i suoi multipli.
- Si passa al numero successivo non eliminato e si ripete l'operazione.
- Quando non rimangono più numeri da "cribrare", quelli non eliminati sono i numeri primi fino a n .

Scrivere un programma in C che, dato un valore n , utilizzi un array allocato dinamicamente per implementare il crivello e stampi tutti i numeri primi minori o uguali a n .

Suggerimento: creare un array dinamico "buffer" di `int` di lunghezza n , i cui valori sono inizializzati a 1 (nessun numero eliminato, escluso 1). Per ogni intero i da 2 a \sqrt{n} utilizzare un ciclo per azzerare valori di buffer in corrispondenza degli indici che sono multipli interi di i e non sono già stati eliminati.