

# Politecnico di Milano

## Informatica B, A.A. 2021/2022

### Laboratorio 3

Luca Frittoli (luca.frittoli@polimi.it)  
Mirko Salaris (mirko.salaris@polimi.it)

17 novembre 2021

1. **Statistiche di base** Si scriva un programma che prenda in input un array di numeri interi e stampi il minimo, massimo, media e somma.
2. **Aiuta gli escursionisti** Per aiutare degli escursionisti, scrivi un programma che accetti una sequenza di  $N$  dati di elevazione. È importante per gli escursionisti poter poi visualizzare la sequenza sia nell'ordine originale sia nella direzione opposta. Chiedi quindi in quale direzione si voglia visualizzare la sequenza e successivamente stampala a video, nella direzione richiesta.  
**Bonus:** il sentiero può essere percorso in entrambe le direzioni e partendo da qualsiasi punto. Riscrivi il programma per supportare questa funzionalità.
3. **Raddoppi** Accetta dall'utente una sequenza di numeri interi positivi che termini con 0 e che sia lunga al massimo 100 numeri. Analizza quindi la sequenza e stampa le coppie di numeri consecutivi che sono uno il doppio del precedente.

Esempio:

```
Sequenza di input: 1 2 7 14 3 6 12 10 11 0
Output:
  1 2
  7 14
  3 6
  6 12
```

4. **Operazioni con matrici** Si scriva un programma che prenda in input 2 matrici di interi di dimensione  $3 \times 3$ , ne calcoli la somma (elemento per elemento) e la stampi a schermo.

**Esempio:**

```
1 2 3    5 1 4    6 3 7
4 5 6 +  5 6 12 =  9 11 18
7 8 9    7 2 11    14 10 20
```

**Bonus:** modificare il programma in modo che venga calcolato il prodotto riga per colonna fra matrici. Ricordarsi di verificare che le dimensioni delle matrici siano compatibili.

5. **Area di un triangolo qualsiasi** Si scriva un programma che prenda in input le coordinate di tre punti nel piano e calcoli l'area del triangolo che abbia i tre punti come vertici.

**Consigli:**

- Salvare le coordinate dei vertici in una matrice  $2 \times 3$  e le lunghezze dei lati in un array permette di evitare di ripetere parti di codice.
  - La formula di Erone permette di calcolare l'area di un triangolo date le lunghezze dei lati  $a, b, c$  e il semiperimetro  $p$  come  $A = \sqrt{p(p-a)(p-b)(p-c)}$ .
  - Controllare la correttezza del risultato ottenuto con triangoli rettangoli con lati paralleli agli assi.
6. **A - Una semplice struttura dati** Dichiarare un tipo di dato `Esame` che contenga il nome del corso e il voto ottenuto nel corso. Scrivere poi un programma che prenda in input un `Esame` e ne stampi i dati.
6. **B - Media dei voti** Dichiarare una struttura dati che usi il tipo di dati `Esame` definito nell'esercizio precedente per descrivere uno `Studente` tramite il nome e dalla lista degli esami sostenuti (3 esami). Scrivere poi un programma che prenda in input uno studente e calcoli la media dei suoi voti.
7. **Tema d'Esame del 25/01/2021 (10 punti)** Giovo vi ha incaricato di creare un programma per la gestione delle consegne a domicilio. Il programma mantiene un vettore di richieste di consegne a domicilio prodotte dai vari clienti (si chiami questa variabile: `richieste`); ciascuna richiesta è caratterizzata dal codice cliente (`codice_cliente`, un intero), indirizzo del cliente (`ind_cli`, codificato per semplicità in termini di due coordinate GPS: `float x` e `y`), indirizzo del commerciante (`ind_commerciante`, anch'esso codificato come due coordinate GPS: `float x` e `y`) e prezzo pagato per la consegna (`costo_merce`). Il programma mantiene inoltre un vettore di fattorini (nome variabile: `fattorini`), caratterizzati ciascuno dal codice fattorino (`codice_fattorino`, un intero), dalla posizione (sempre in termini di due coordinate GPS: `float x` e `y`) e da una variabile logica `occupato` che indica se il fattorino sta già effettuando una consegna oppure no.
1. Si definiscano in linguaggio C i tipi necessari a rappresentare richieste e fattorini. Si dichiarino, inoltre, le variabili array `richieste` e `fattorini` dimensionandole tramite due costanti definite con nome `N_RICHIESTE` e `N_FATTORINI`, rispettivamente.
  2. Si scriva una porzione di codice in linguaggio C che, partendo dai due vettori `fattorini` e `richieste` completamente popolati, per ogni richiesta, ricavi l'indice del fattorino libero più vicino al commerciante (a tale scopo, si calcoli la distanza Euclidea tra fattorino e commerciante). La porzione di codice applica, inoltre, uno sconto del 20% al costo merce se la distanza tra commerciante e fattorino è inferiore alla distanza fra commerciante e cliente (si modifichi il campo `costo_merce` della richiesta per applicare lo sconto). Per ogni richiesta, si stampino l'indice della richiesta, le coordinate del cliente, del commerciante e del fattorino, insieme al costo eventualmente scontato.
- Suggerimento:** La distanza euclidea tra due punti in  $(x_1, y_1)$  e  $(x_2, y_2)$  è definita nel seguente modo:  
 $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ .
8. **Autosalone** Definire una struttura che descriva un Modello di automobile in termini di marchio, nome del modello e cilindrata (per esempio `marchio = 'Fiat'`, `nome = 'Panda'`, `cilindrata = 1000`), e una struttura che descriva un'Automobile in termini di Modello, cognome del proprietario e anno di immatricolazione. Scrivere poi un programma che permetta all'utente di inserire i dati di un numero a scelta di automobili di un autosalone (compreso tra 1 e 10) e stampi a video:
1. il numero di automobili di un marchio (scelto dall'utente) immatricolati a partire dal 2019, e
  2. i cognomi dei proprietari di automobili di cilindrata superiore a 1800.
9. **Maiuscole/minuscole** Si scriva un programma che prenda in input una stringa e converta i caratteri minuscoli in maiuscoli e vice-versa.
- Nota:** per acquisire una stringa di testo utilizza la funzione `fgets(variable_stringa, max_caratteri, stdin)`, dove `stdin` va scritto esattamente così.
- Esempio:

```
char testo[240];
fgets(testo, 240, stdin);
```

10. **Suddivisione in parole** Nelle applicazioni di NLP (processazione del linguaggio “umano”) spesso è richiesto lavorare sulle singole parole. Come esercizio, scrivi quindi un programma in grado di separare le singole parole all’interno di una frase. Il tuo programma deve accettare in input una stringa di testo di qualsiasi lunghezza e restituire in output la lista delle parole in essa contenuta, andando a capo ad ogni parola.

**Consiglio:** il programma completo dovrebbe gestire anche la punteggiatura (eliminandola), però prova ad iniziare considerando testo composto solo da lettere e spazi.

## Esercizi extra

- Extra 1. **Parentesi** Come esercizio introduttivo alla programmazione di una calcolatrice scientifica, scrivi un programma che accetti in input una stringa contenente parentesi tonde e verifichi che il numero di parentesi aperte e chiuse combaci.

Esempi:

```
()()    -> Si
(())    -> Si
)))(((  -> Si
())     -> No
```

**Bonus:** restituisci un responso positivo solo se le parentesi sono correttamente innestate.

Esempi:

```
()()    -> Si
(())    -> Si
)))(((  -> No
())     -> No
```

- Extra 2. **Cifrario di Cesare** Si scriva un programma che prenda in input una stringa (plaintext) senza spazi e fatta solo di caratteri maiuscoli, e un numero intero (key). Il programma cifra la stringa plaintext con il Cifrario di Cesare con chiave key, stampando il risultato cifrato (cyphertext) a schermo.

Il Cifrario di Cesare è un semplice cifrario a rotazione, che sostituisce una lettera dell’alfabeto con un’altra, ruotandole di key posti.

**Esempio:**

```
key = 3
plaintext = A B C D E F G ... X Y Z
cyphertext = D E F G H I J ... A B C
```

**Bonus:** Scambia un messaggio cifrato, e la chiave, con un tuo collega e provate a decifrarlo (suggerimento: il programma accetta anche chiavi negative).