

Introduzione

In questo documento potrete trovare alcuni esercizi aggiuntivi rispetto a quelli visti durante le esercitazioni. Gli esercizi saranno divisi per sezioni in base agli argomenti trattati durante le lezioni frontali. In generale, a meno che non venga espressamente specificato dal testo dell'esercizio, **non è consentito l'utilizzo di funzioni di libreria particolari**. Ad esempio, l'utilizzo di `<iostream>` è consentito, mentre l'utilizzo di `<cmath>` è consentito solo in casi speciali.

Questo documento verrà aggiornato durante il corso con del nuovo materiale, seguendo l'ordine degli argomenti visti a lezione. Se avete bisogno di delucidazioni per quanto riguarda gli esercizi, utilizzate pure la funzione forum della Didattica Online, oppure scrivete una mail a noi esercitatori.

Giovanni De Toni (giovanni.detoni@unitn.it)
Stefano Berlato (stefano.berlato-1@unitn.it)

1 Esercizi base

1. Scrivere un programma che chiede all'utente di inserire un valore (numero o carattere). Salvare questo valore in una variabile e stamparlo poi a video.
2. Scrivere un programma in C++ che stampi a video la grandezza in byte dei diversi tipi base. **Nota: in questo caso, è consentito l'utilizzo della funzione `sizeof()`.**
3. Scrivere un programma che ritorni a video la somma di due numeri;
4. Scrivere un programma che ritorni a video il risultato delle operazioni aritmetiche (+, -, *, etc.) tra variabili di tipo diverso (e.g., float, int, etc.).
5. Scrivere un programma che, ricevuto in input dall'utente un valore di distanza in miglia, ritorni lo stesso valore convertito in chilometri.

$$d_{km} = \frac{d_{miglia}}{0.62137} \quad (1)$$

6. Scrivere un programma che, data in input la lunghezza di una lato L , calcoli l'area e il perimetro di un esagono. **Nota: per calcolare $\sqrt{3} = 1.73205$, potete definire una variabile costante e assegnarle direttamente il valore. Volendo, si può anche utilizzare la funzione `sqrt` della libreria `<cmath>`;**

$$P = 6 \cdot L \quad A = 3 \frac{L^2}{2} \sqrt{3} \quad (2)$$

7. Scrivere un programma che determini se il computer che state utilizzando lavora in Big Endian o Little Endian.

2 Esercizi Istruzioni Condizionali

1. Scrivere un programma che, dati in input tre valori dall'utente, riporti a video il maggiore di questi tre valori (interi, float, caratteri, etc.);
2. Scrivere un programma che, data una temperatura in input dall'utente, riporti a video un messaggio seguendo le regole sotto elencate:

$$\text{message}(T) = \begin{cases} T < -273.15 & \text{Temperatura impossibile!} \\ -273.15 \geq T \geq 0.0 & \text{Fa freddo!} \\ 0.0 > T \geq 18.0 & \text{Temperatura così così} \\ 18.0 > T \geq 30.0 & \text{Temperatura accettabile} \\ T > 30.0 & \text{Fa caldo!} \end{cases} \quad (3)$$

3. Scrivere un programma che, dati in input i lati di un triangolo, ritorni a video se il triangolo è isoscele, equilatero o scaleno;
4. Scrivere un programma che, dato in input il valore numerico di un mese, ritorni a video il corrispondente nome di quel mese (e.g., 1-Gennaio, ..., 12-Dicembre);

3 Esercizi Istruzioni Iterative

1. Scrivere un programma che, dato in input un valore N , ritorni a video il risultato della seguente serie geometrica, calcolata fino all' N -esimo termine.

$$\sum_{i=0}^N \frac{1}{2^i} = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = \sum_{i=1}^{\infty} = 2 \quad (4)$$

2. Scrivere un programma che, dato in input un valore N , ritorni a video il risultato della seguente serie che approssima il valore di π .

$$\sum_{i=1}^N \frac{(-1)^{1+i} 4}{2i-1} = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \dots = \sum_{i=0}^{\infty} \frac{(-1)^{1+i} 4}{2i-1} = \pi \quad (5)$$

3. Scrivere un programma che, dato in input il valore x e il valore N , calcoli l'espansione fino all' N -esimo termine della serie di Taylor calcolata per e^x . La formula esatta equivale a:

$$e^x = \sum_{i=0}^N \frac{x^i}{i!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \quad (6)$$

Il termine $x!$ corrisponde al fattoriale del valore x .

4. Scrivere un programma che, dato in input un numero intero N , ritorni a video la somma delle sue cifre. Ad esempio, dato in input il numero 13465, l'output del programma dovrà essere 19, poichè $1 + 3 + 4 + 6 + 5 = 19$.
5. Scrivere un programma che, dato in input un numero intero N , ritorni a video la frequenza di ogni cifra (tra 0 e 9) contenuta nel numero. Ad esempio, dato in input il numero 1231156, l'output sarà:

(7)

```

Frequenza 0 : 0
Frequenza 1 : 3
Frequenza 2 : 1
Frequenza 3 : 1
Frequenza 4 : 0
... e via dicendo fino al 9

```

6. Scrivere un programma che, dato un numero N , ritorni a video la scomposizione in fattori primi di questo numero. Ad esempio, dato in input il numero 22, il programma dovrà ritornare in output 2, 11.
7. Scrivere un programma che, dato in input un numero binario N , ritorni a video il complemento di questo numero binario. Ad esempio, dato in input il numero 110101, il programma dovrà dare in output il numero 001010.
8. Scrivere un programma che, dato in input un intero N , stampi a video il numero scritto però con parole corrispondenti alle cifre. Ad esempio, dato in input il numero 134, l'output del programma dovrà essere "Uno Tre Quattro".
9. Scrivere un programma che, generata una lettera dell'alfabeto (maiuscola o minuscola) in modo casuale, chieda all'utente di indovinare suddetta lettera. Il programma terminerà solo se l'utente avrà indovinato. **Nota: in questo caso, è consentito l'utilizzo della funzione `rand()` della libreria `<stdlib>`.**
10. Scrivere un programma che, dato in input un numero N inserito dall'utente, stampi a video un rombo, in cui ogni riga deve contenere un numero dispari di asterischi. Il numero di righe del rombo è indicato dalla formula $2N - 1$.

N = 3

```

  *
 ***
*****
 ***
  *

```

4 Esercizi sulla Ricorsione

1. Scrivere un programma che prenda in input due interi (riga e colonna) e ritorni il corrispondente numero del Triangolo di Pascal tramite funzione ricorsiva.

						1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
--	--	--	--	--	--	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

2. Scrivere un programma che prenda in input un intero (riga) e stampi il Triangolo di Pascal fino alla riga specificata tramite funzione ricorsiva.
3. Scrivere un programma che prenda in input due interi (base ed esponente) e calcoli la potenza tramite funzione ricorsiva.
4. Scrivere un programma che prenda in input un intero e calcoli la somma delle cifre dell'intero tramite funzione ricorsiva.
5. Scrivere un programma che prenda in input un intero e controlli se l'intero è palindromo tramite funzione ricorsiva.
6. Scrivere un programma che prenda in input un intero e stampi a video la sequenza di Hailstone¹ tramite funzione ricorsiva.

$$f(n) = \begin{cases} 1 & \text{se } n = 1 \\ n/2 & \text{Se } n \text{ è pari} \\ 3n + 1 & \text{Se } n \text{ è dispari} \end{cases} \quad (8)$$

7. Scrivere un programma con una funziona ricorsiva che si invoca fino ad riempire lo stack e ottenere un errore Segmentation Fault. Stampare a video la profondità raggiunta usando un contatore. Usare `cout.flush();`² per essere sicuri di stampare a video il contatore prima che il processo termini per Segmentation Fault (sul computer dell'esercitatore il numero è 261991).

5 Esercizi su Array

1. Scrivere un programma che ruoti un array di n elementi di X posizioni. Una rotazione corrisponde a spostare l'elemento in indice i in indice i+1% n.
2. Scrivere un programma che controlli se un dato array è un sottovettore di un altro dato array.
3. Scrivere un programma che implementi la ricerca sequenziale di un elemento in un array tramite ricorsione.
4. Scrivere un programma che implementi l'algoritmo di ordinamento Quicksort.
5. Scrivere un programma che implementi l'algoritmo di ordinamento Merge Sort.
6. Scrivere un programma che implementi l'algoritmo di ordinamento Random Sort (scegliere due indici a caso, scambiare gli elementi e controllare se l'array è ordinato oppure no. Ripetere fino a che l'array non è ordinato).
7. Scrivere un programma che dichiari e poi inizializzi (con valori random) un array di 10 interi, lo stampi a video, lo inverta tramite funzione ricorsiva e poi stampi l'array invertito. Si può utilizzare la funzione `rand` della libreria `<cstdlib>`.
8. Scrivere un programma che prenda in input un array di 5 caratteri e ne stampi a video la codifica di Cesare (sostituire ad ogni lettera la corrispondente lettera tre posizioni successiva nell'alfabeto. Tutti gli altri caratteri rimangono inalterati). Si può utilizzare la funzione `rand` della libreria `<cstdlib>`.

¹https://it.wikipedia.org/wiki/Congettura_di_Collatz

²<https://en.cppreference.com/w/cpp/io/manip/flush>

['C', 'i', 'a', 'o', '!'] => ['F', 'l', 'd', 'r', '!']

9. Scrivere un programma che dichiari e poi inizializzi (con valori random) un array di 10 interi, lo stampi a video e conti il numero totale di elementi duplicati nell'array. Si può utilizzare la funzione rand della libreria <stdlib>.
10. Scrivere un programma che dichiari e poi inizializzi (con valori random) un array di 10 interi, lo stampi a video e conti la frequenza di ogni elemento nell'array. Si può utilizzare la funzione rand della libreria <stdlib>.
11. Scrivere un programma che dichiari e poi inizializzi (con valori random) un array di 10 interi, lo stampi a video e poi sposti tutti gli zeri alla fine dell'array. Si può utilizzare la funzione rand della libreria <stdlib>.
12. Scrivere un programma che dichiari e poi inizializzi (con valori random) due matrici di 10x10 interi e ne calcoli la somma, sottrazione e prodotto. Si può utilizzare la funzione rand della libreria <stdlib>.
13. Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 10x10 interi e ne calcoli il determinante tramite il Teorema di Laplace³. Si può utilizzare la funzione rand della libreria <stdlib>.
14. Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 10x10 interi con valore 0 oppure 1. Scrivere poi una funzione che torni il numero di "isole", definite come elementi della matrice con valore 1 circondati da soli zeri (nell'immagine proposta come esempio qui sotto questo esercizio, ci sono 2 isole). Si può utilizzare la funzione rand della libreria <stdlib>.

1	0	1	0
0	0	0	1
0	1	0	0
1	0	0	1

→

1	0	1	0
0	0	0	1
0	1	0	0
1	0	0	1

15. Scrivere un programma che dichiari e poi inizializzi (con valori random) una matrice di 10x10 interi con valore 0 oppure 1. Scrivere poi una funzione che torni il numero di "isole", definite come gruppi di elementi della matrice con valore 1 circondati da soli zeri (nell'immagine proposta come esempio qui sotto questo esercizio, ci sono 4 isole). Si può utilizzare la funzione rand della libreria <stdlib>.

1	0	1	0
0	0	0	1
0	1	0	0
1	0	0	1

→

1	0	1	0
0	0	0	1
0	1	0	0
1	0	0	1

16. Implementare a piacimento il gioco Tris⁴ usando le matrici.
17. Implementare a piacimento il gioco Battaglia Navale⁵ usando le matrici.

³https://it.wikipedia.org/wiki/Teorema_di_Laplace

⁴[https://it.wikipedia.org/wiki/Tris_\(gioco\)](https://it.wikipedia.org/wiki/Tris_(gioco))

⁵[https://it.wikipedia.org/wiki/Battaglia_navale_\(gioco\)](https://it.wikipedia.org/wiki/Battaglia_navale_(gioco))

6 Esercizi su Liste Concatenate

1. Scrivere un programma che, dato in input un file con un intero per riga, crei, stampi ed elimini una lista concatenata con i numeri contenuti nel file.
2. Scrivere un programma che, dato in input un file con un intero per riga, crei, stampi ed elimini una doppia lista concatenata con i numeri contenuti nel file.
3. Scrivere un programma che, data una lista concatenata di interi, riceva dall'utente un intero in input e cerchi quell'intero nella lista. Se presente, il programma deve eliminare il nodo corrispondente.
4. Scrivere un programma che, data una lista concatenata di interi, riceva dall'utente due indici ed inverta i nodi corrispondenti nella lista. Se un indice è maggiore del numero degli elementi presenti nella lista, ritornare un errore all'utente.
5. Scrivere un programma che, dato un array di interi, lo converta in una lista concatenata.
6. Scrivere un programma che, data una lista concatenata di interi, la converta in un array.
7. Scrivere un programma che, data una lista concatenata di interi, scorra la lista ed elimini i nodi duplicati (cioè con lo stesso intero).
8. Scrivere un programma che, data una lista concatenata di interi in ordine crescente, chieda in input all'utente un nuovo intero e lo inserisca in modo ordinato nella lista.
9. Scrivere un programma che, data una lista concatenata di interi in ordine crescente, chieda in input all'utente un nuovo intero e lo inserisca in modo ordinato nella lista tramite ricorsione.
10. Scrivere un programma che, date due liste concatenate di interi in ordine crescente, faccia il merge ordinato in una terza lista.
11. Scrivere un programma che, data una lista concatenata di interi, inverta l'ordine degli elementi della lista (il primo elemento diventa l'ultimo, il secondo diventa il penultimo, etc.).
12. Scrivere un programma che, data una lista concatenata di interi, inverta la lista.
13. Scrivere un programma che, data una lista concatenata di interi, ritorni una seconda lista concatenata di interi cosicchè questa seconda lista contenga l'elemento maggiore successivo più vicino per ogni elemento nella lista di partenza (0 se nessun elemento è maggiore).

`[1,8,2,7,9,3] => [8,9,7,9,0,0]`

7 Esercizi su Pile e Code

1. Implementare una pila tramite array dinamici. Ricordarsi di aggiornare la dimensione dell'array in caso sia raggiunto il limite degli elementi (i.e., creare un array più grande e copiarci gli elementi).
2. Implementare una coda tramite array dinamici. Ricordarsi di aggiornare la dimensione dell'array in caso sia raggiunto il limite degli elementi (i.e., creare un array più grande e copiarci gli elementi).
3. Implementare una pila tramite liste concatenate.
4. Implementare una coda tramite liste concatenate.
5. Implementare una coda usando due stack.
6. Scrivere un programma che riceva in input un'espressione matematica di interi con un'unica operazione disponibile, cioè la somma. Eseguire poi la somma.

`1 + 4 => 5`

7. Arricchire il programma precedente gestendo anche la moltiplicazione. Ricordarsi di gestire la precedenza.

$$1 + 4 * 2 \Rightarrow 9$$

8. Scrivere un programma che, data una matrice (labirinto), una cella di partenza e una cella di arrivo, restituisca 1 se è possibile raggiungere l'arrivo dalla partenza passando solo per celle del labirinto di valore 0. Non ci si può muovere in diagonale. Le celle con valore 1 rappresentano i muri (hint: utilizzare una pila).