

# C++ #10 Esercizi con le strutture

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo delle strutture.

Gli esercizi, a complessità crescente, possono essere risolti utilizzando anche differenti linguaggi di programmazione come C, Python, Pascal, Java, PHP.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

## ESERCIZI CON STRUTTURE

1) Realizzare tre differenti strutture per: modellare un rettangolo, un quadrato ed un cerchio. Utilizzare queste strutture per permettere ad un utente di inserire base ed altezza del rettangolo, lato del quadrato e raggio del cerchio. Calcolata l'area delle differenti forme geometriche, il programma deve riportare in seguito quale delle tre è caratterizzata dall'area maggiore.

2) Realizzare un programma che preveda l'utilizzo di una struttura auto. La struttura permette di descrivere un'automobile in base a marca, cilindrata, anno di immatricolazione e acquirente. Il programma deve prevedere

inoltre l'utilizzo di un Array di tipo auto (con dimensioni pari a 5) che consenta di inserire da tastiera i dati delle auto in vendita in un autosalone. (N.B. si utilizzi un ciclo definito per iterare su tutti gli elementi dell'array)

3) Realizzare un programma che preveda l'utilizzo di una struttura libro. La struttura permette di descrivere un libro in base al codice ISBN (alfanumerico), numero di pagine, prezzo, anno di pubblicazione autore. Il programma deve prevedere inoltre prevedere la memorizzazione di tre differenti libri e, dopo avere inserito i dati, riportare i tre libri in ordine crescente per anno di pubblicazione (dal più vecchio al più nuovo).

---

## C++ #9 Scrittura e Lettura da File

**Obiettivo:** Gestire la lettura e la scrittura su file utilizzando le relative funzioni c++.

### Teoria:

Il fatto che un file abbia una vita indipendente dal programma utilizzato per la sua creazione/modifica/lettura permette l'implementazione di algoritmi complessi che prevedono l'impiego della memoria permanente e non solo della memoria volatile (tipicamente impiegata per l'elaborazione dei dati). Le operazioni tipicamente effettuate da un programma su uno specifico file sono 4. Nel dettaglio:

- **Apertura:** Il programma comunica al sistema operativo la necessità di accedere ad uno specifico file.
- **Chiusura:** Il programma comunica al sistema operativo che non ha più necessità di accedere ad uno specifico file.
- **Apertura:** Il programma richiede dei dati dal file come input.
- **Chiusura:** Il programma intende scrivere dei dati in un file come output.

Attenzione: Un programma non può leggere o scrivere dei dati se prima non ha aperto il file in questione.

La libreria tipicamente utilizzata per la gestione dei file è la `fstream` (file con accesso contemporaneo sia in lettura che in scrittura).

**Codice:**

---

# C++ #8 Esercizi con le Matrici

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo delle matrici.

Gli esercizi, a complessità crescente, possono essere risolti utilizzando anche differenti linguaggi di programmazione come C, Python, Pascal, Java, PHP.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

## ESERCIZI CON MATRICI (DIFFICOLTA EASY)

1) Realizzare un programma che permetta di inserire in tutte le posizioni di un matrice di dimensione 5×5 un numero a scelta introdotto dall'utente. (N.B. In tutte le celle deve essere presente lo stesso numero)

2) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4. (N.B. In tutte le celle deve essere presente un numero diverso scelto dall'utente)

2 bis) Realizzare un programma che permetta di riempire,

attraverso numeri scelti dall'utente, una matrice di dimensione 4x4 e stamparne in seguito il contenuto.

**3)** Realizzare un programma che permetta di riempire, attraverso numeri casuali compresi tra 0 e 9, una matrice di dimensione 4x4 e stamparne in seguito il contenuto.

**4)** Realizzare un programma che permetta di riempire, attraverso numeri casuali compresi tra 0 e 9, una matrice di dimensione 4x4. Determinare in seguito se un numero a scelta introdotto dall'utente è presente in una qualsiasi cella di una matrice 4x4.

**4 bis)** Realizzare un programma che permetta di determinare quante volte un numero a scelta, introdotto dall'utente, è presente in una qualsiasi cella della matrice 4x4.

Esempio

**Inserire un numero: 3**

**Il numero 3 è presente 2 volte nella matrice**

3	2	1	6
4	3	1	5
6	4	2	6
4	1	4	7

**5)** Realizzare un programma che permetta di riempire, con tutti zeri tranne nelle celle della diagonale principale (dove deve

memorizzare uno), una matrice di dimensione 4×4 e stamparne in seguito il contenuto.

Esempio

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

### **ESERCIZI CON MATRICI (DIFFICOLTA MEDIUM)**

6) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4 e ne calcoli la somma di tutti gli elementi presenti.

7) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4 e dopo avere determinato la somma di tutti gli elementi presenti per ogni riga, determini la riga con la somma maggiore.

8) Realizzare un programma che permetta di riempire, attraverso numeri scelti dall'utente, una matrice di dimensione 4×4. In seguito determini se tutti gli elementi presenti nella matrice risultino maggiori di n e minori di m (con n ed m forniti dall'utente).

9) Realizzare un programma che permetta di memorizzare in una matrice 10×10 la tavola pitagorica

---

# C++ #7 Esercizi con gli Array

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo degli array.

Gli esercizi, a complessità crescente, possono essere risolti utilizzando anche differenti linguaggi di programmazione come C, Python, Pascal, Java, PHP.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

## ESERCIZI CON ARRAY (DIFFICOLTA' EASY)

1) Realizzare un programma che inserisca automaticamente nelle prime dieci posizioni di un array i primi dieci numeri naturali (e.g., 0, 1, 2, ... , 9).

2) Realizzare un programma che memorizzi nelle prime dieci posizioni di un array, dieci differenti numeri forniti come input da un utente.

3) Realizzare un programma che inserisca automaticamente nelle prime dieci posizioni di un array i primi dieci numeri

naturali riportati in ordine inverso (e.g., 9, 8, 7, ... , 1).

### **ESERCIZI CON ARRAY E BLOCCHI CONDIZIONALI (DIFFICOLTA' MEDIUM)**

**4)** Realizzare un programma che determini se in un array di dimensione 5 è presente il valore 0. (**N.B.** prima di eseguire la ricerca l'utilizzatore deve inserire dei valori attraverso la procedura presentata nell'esercizio 2).

**5)** Realizzare un programma che determini il numero più grande presente in un array di dimensione 5. (**N.B.** prima di eseguire la ricerca l'utilizzatore deve inserire dei valori attraverso la procedura presentata nell'esercizio 2).

**5bis)** Realizzare un programma che determini sia il numero più grande sia il numero più piccolo presente in un array di dimensione 5.

**6)** Realizzare un programma che determini la somma dei numeri presenti in un array di dimensione 5. (**N.B.** prima di eseguire la ricerca l'utilizzatore deve inserire dei valori attraverso la procedura presentata nell'esercizio 2).

**6bis)** Realizzare un programma che determini la media dei numeri presenti in un array di dimensione 5.

**8)** Realizzare un programma che, dato un array di dimensione 10, inserisca automaticamente zero nelle posizioni caratterizzate da un indice pari ed un in quelle caratterizzate da un indice dispari (**N.B.** l'array deve

contenere i seguenti numeri 0, 1, 0, 1, 0, 1, 0, 1, 0, 1).

### ESERCIZI CON ARRAY E BLOCCHI CONDIZIONALI (DIFFICOLTA': HARD)

**9)** Realizzare un programma che determini il numero più frequente in un array di dimensione 5. (N.B. prima di eseguire la ricerca l'utilizzatore deve inserire dei valori attraverso la procedura presentata nell'esercizio 2).

**10)** Realizzare un programma che determini se il contenuto di un array di caratteri dimensione 10 risulta palindromo. Le parole mantengono lo stesso suono e lo stesso significato sia che si leggano normalmente, da sinistra verso destra, sia viceversa, cioè da destra verso sinistra. (N.B. prima di eseguire l'operazione richiesta l'utilizzatore deve inserire i caratteri nell'array).

---

## **C++ #6 Esercizi con le funzioni**

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo delle funzioni.

Gli esercizi, a complessità crescente, prevedono inizialmente la realizzazione di funzioni con passaggio dei parametri per valore ed in seguito per riferimento.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

### **FUNZIONI CON PASSAGGIO PARAMETRI PER VALORE**

1) Implementare una funzione che ricevuto un numero in ingresso restituisca il valore assoluto di tale numero. Realizzare un programma che utilizzi tale funzione.

2) Implementare una funzione che ricevuti due numeri in ingresso restituisca il numero maggiore. Realizzare un programma che utilizzi tale funzione.

3) Implementare una funzione che ricevuti in ingresso i valori della base e dell'altezza di un rettangolo restituisca il valore dell'area. Realizzare un programma che utilizzi tale funzione.

4) Implementare una funzione che ricevuti in ingresso i valori della base e dell'altezza di un rettangolo restituisca il valore del perimetro. Realizzare un programma che utilizzi tale funzione.

5) Implementare una funzione che ricevuto in ingresso un carattere restituisca il valore 0 se il carattere ricevuto è una consonante mentre restituisca 1 se il carattere ricevuto è una vocale. Realizzare un programma che utilizzi tale funzione.

6) Implementare una funzione che ricevuto un numero in ingresso restituisca il valore del fattoriale del numero ricevuto. Realizzare un programma che utilizzi tale funzione. Nota bene, il fattoriale di un numero  $N$  è  $N*(N-1)*(N-2)...*1$

7) Implementare una funzione che ricevuti in ingresso due interi, calcoli il Minimo Comune Multiplo e restituisca tale valore. Realizzare un programma che utilizzi tale funzione.

8) Implementare una funzione che ricevuti in ingresso due interi, calcoli il Massimo Comune Divisore e restituisca tale valore. Realizzare un programma che utilizzi tale funzione.

9) Implementare una funzione che ricevuto in ingresso un numero intero, restituisca true se è un numero primo, altrimenti restituisca false. Realizzare un programma che utilizzi tale funzione.

### **FUNZIONI CON PASSAGGIO PARAMETRI PER RIFERIMENTO**

10) Implementare una funzione che ricevute due variabili in ingresso permetta di scambiare il valore di tale variabili. Realizzare un programma che utilizzi tale funzione.

11) Implementare una funzione che ricevuto in ingresso il valore della base e dell'altezza di un rettangolo permetta di calcolare area e perimetro. Questi ultimi due valori devono essere passati sfruttando la tecnica del riferimento. Realizzare un programma che utilizzi tale funzione.

**12)** Implementare una funzione che ricevuto in ingresso il valore dei parametri  $a$ ,  $b$ , e  $c$  di una equazione di secondo grado  $ax^2+bx+c=0$  calcoli le soluzioni  $x_1$  e  $x_2$  della suddetta equazione . Questi ultimi due valori devono essere passati sfruttando la tecnica del riferimento. Realizzare un programma che utilizzi tale funzione.

**13)** Implementare una funzione che ricevuti in ingresso il valore del numeratore e denominatore di una frazione, riduce la frazione ai minimi termini. Realizzare un programma che utilizzi tale funzione.

**14)** Implementare una funzione che ricevuti in ingresso il valore del numeratore e denominatore di due differenti frazioni, calcoli numeratore e denominatore della frazione ottenuta sommando le due precedenti frazioni. Questi ultimi due valori devono essere passati sfruttando la tecnica del riferimento. Realizzare un programma che utilizzi tale funzione.

## **FUNZIONI CON PASSAGGIO DI ARRAY**

**15)** Implementare una funzione che ricevuto in ingresso un array e la sua dimensione inserisca al suo interno tutti zero. Realizzare un programma che utilizzi tale funzione.

**16)** Implementare due differenti funzioni. Una prima funzione che ricevuti in ingresso un array e la sua dimensione, inserisca al suo interno valori scelti dall'utente. Una seconda funzione che ricevuti in ingresso un array e la sua dimensione riporti i valori contenuti nell'array. Realizzare

un programma che utilizzi tali funzioni di input e output.

**17)** Implementare una funzione che riceve in ingresso un array e la sua dimensione calcoli il valore massimo presente all'interno dell'array e lo restituisca. Realizzare un programma che utilizzi tale funzione.

**18)** Implementare una funzione che riceve in ingresso un array, la sua dimensione, ed un intero N, calcoli quante volte il valore N è presente all'interno dell'array e lo restituisca. Realizzare un programma che utilizzi tale funzione.

---

## **C++ #5 Esercizi con numeri random**

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo dei numeri randomici (numeri casuali).

Gli esercizi, a complessità crescente, prevedono inizialmente la semplice generazione di numeri random ed infine applicazioni più complesse che richiedono l'impiego di cicli iterativi e blocchi condizionali.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

### **ESERCIZI CON NUMERI RANDOMICI**

1) Realizzare un programma che simuli il lancio di un dado a sei facce e riporti il risultato.

2) Realizzare un programma che simuli il lancio di una moneta e riporti il risultato (testa o croce).

3) Realizzare un programma che simuli il lancio di un dado a 20 facce (utile per i giocatori di Dungeons and Dragons) e riporti il risultato.

4) Realizzare un programma che permetta all'utente di definire il numero di facce di un dado e ne simuli il lancio riportando il risultato.

### **ESERCIZI CON NUMERI RANDOMICI E BLOCCHI CONDIZIONALI**

5) Realizzare un programma che simuli il lancio di due dadi a sei facce e dopo avere riportato il risultato indichi se è uscita una coppia di numeri uguali.

6) Realizzare un programma che permetta di simulare l'estrazione del lotto attraverso la generazione di 5 numeri casuali nel range 1-90. Attenzione i numeri non possono essere ripetuti e devono risultare equiprobabili.

## ESERCIZI CON NUMERI RANDOMICI E CICLI ITERATIVI

7) Realizzare un programma che generato un numero naturale nel range 0-10, permetta all'utente di indovinare il numero generato. Il programma deve terminare quando il numero viene indovinato (CICLO INDEFINITO).

**7bis)** Realizzare un programma che generato un numero naturale nel range 0-10, permetta all'utente di indovinare il numero generato. Il programma deve terminare quando il numero viene indovinato (CICLO INDEFINITO). Ad ogni iterazione, nel caso in cui l'utente non abbia individuato, deve essere fornito un feedback all'utilizzatore in cui si riporta se il numero generato risulta maggiore o inferiore al numero inserito.

---

# C++ #4 Esercizi con cicli iterativi e blocchi condizionali

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo di cicli iterativi e blocchi condizionali.

Gli esercizi, a complessità crescente, devono essere risolti utilizzando cicli iterativi (definiti e indefiniti) . L'utente può scegliere in modo autonomo se utilizzare i costrutti while, do-while, for e se in caso impiegare blocchi condizionali semplici, nidificati, o se sfruttare la logica

booleana.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

### **ESERCIZI CON CICLI ITERATIVI E BLOCCHI CONDIZIONALI**

1) Realizzare un programma che permetta all'utente di inserire 10 numeri naturali e determini quanti numeri positivi e negativi siano stati inseriti.

2) Realizzare un programma che permetta all'utente di inserire N numeri naturali e determini quanti numeri positivi e negativi siano stati inseriti. Il programma deve terminare quando l'utente inserisce un numero uguale a 0.

3) Realizzare un programma che permetta all'utente di inserire 10 numeri naturali e determini quanti numeri pari e dispari siano stati inseriti. Utilizzare l'operatore modulo per determinare se un numero è pari oppure dispari.

4) Realizzare un programma che permetta all'utente di inserire 5 voti e ne determini la media, il valore massimo ed il valore minimo.

**4bis)** Realizzare un programma che permetta all'utente di inserire 5 voti e ne determini la media, il valore massimo ed

il valore minimo. Inserire inoltre un controllo per fare in modo che nel caso in cui sia inserito un voto non valido (maggiore di 10 o minore di 0) il programma richieda di inserire nuovamente il dato.

**5)** Realizzare un programma che permetta all'utente di inserire 5 voti e per ogni voto indichi se questo risulta sufficiente oppure insufficiente.

**6 NEW)** Realizzare un programma che permetta all'utente di inserire una sequenza di numeri interi, terminata dall'inserimento del numero zero. Il programma dovrebbe quindi determinare se la sequenza inserita è crescente oppure no. Il programma dovrebbe quindi stampare un messaggio appropriato che indichi il tipo di sequenza inserita.

**6bis)** Realizzare un programma che permetta all'utente di inserire una sequenza di numeri interi, terminata dall'inserimento del numero zero. Il programma dovrebbe quindi determinare se la sequenza inserita è decrescente oppure no. Il programma dovrebbe quindi stampare un messaggio appropriato che indichi il tipo di sequenza inserita.

**7 NEW)** Realizzare un programma che permetta all'utente di calcolare il fattoriale di un numero inserito dall'utente (nel caso di un numero negativo viene stampato un messaggio di errore perché il fattoriale non è definito per numeri negativi).

A seguire sono riportati alcuni esempi relativi al calcolo del fattoriale: Il fattoriale di 5 è pari a 120 e si calcola in questo modo  $5*4*3*2*1$ . Il fattoriale di 3 è pari a 6 e si calcola in questo modo  $3*2*1$ . Il fattoriale di 0 è uguale ad 1

---

# C++ #2 Esercizi con cicli iterativi

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo dei blocchi condizionali.

Gli esercizi, a complessità crescente, possono essere risolti utilizzando cicli iterativi definiti e indefiniti. L'utente può scegliere in modo autonomo se utilizzare i costrutti while, do-while e for.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

## ESERCIZI CON ITERAZIONE DEFINITA

1) Realizzare un programma che permetta di scrivere a monitor i primi 100 numeri naturali.

2) Realizzare un programma che, definito un numero N inserito dall'utente, permetta di scrivere a monitor i numeri naturali compresi tra 1 ed N.

3) Realizzare un programma che permetta all'utente di inserire 10 differenti numeri e ne calcoli la somma.

4) Realizzare un programma che permetta all'utente di inserire 10 differenti numeri e ne calcoli la media.

5) Realizzare un programma che, definito un numero  $N$  inserito dall'utente, permetta di scrivere a monitor i primi 10 multipli del numero inserito (e.g.,  $N$ ,  $2N$ ,  $3N$ , ...,  $10N$ ).

6) Realizzare un programma che permetta di scrivere a monitor i primi  $N$  numeri della successione di Fibonacci. Nel dettaglio, i primi due numeri della successione i Fibonacci sono 0 e 1, i successivi sono dati dalla somma dei due numeri precedenti. Ad esempio: 0, 1, 1, 2, 3, 5, 8, 13 ...

### **ESERCIZI CON ITERAZIONE INDEFINITA**

7) Realizzare un programma che permetta all'utente di inserire una serie indefinita di numeri interi. Il programma deve fermarsi quando la somma dei numeri inseriti supera 100.

9) Realizzare un programma che permetta all'utente di inserire una serie indefinita di numeri interi e ne calcoli la media. Il programma deve fermarsi quando l'utente inserisce uno zero.

10) La popolazione di rane in uno stagno raddoppia ogni giorno. Definita la popolazione iniziale e la popolazione finale da raggiungere, determinare il numero di giorni necessari per raggiungere la popolazione obiettivo.

**10bis)** La popolazione di rane in uno stagno raddoppia ogni giorno. Ogni tre giorni a causa di una epidemia generica la popolazione di rane diminuisce di un terzo. Definita la popolazione iniziale e la popolazione finale da raggiungere, determinare il numero di giorni necessari per raggiungere la popolazione obiettivo.

---

## **C++ #1 Esercizi con blocchi condizionali**

**Obiettivo:** Sono riportati in seguito una serie di esercizi utili per testare le proprie conoscenze nell'ambito dei linguaggi di programmazione strutturata relativamente all'utilizzo dei blocchi condizionali.

Gli esercizi, a complessità crescente, possono essere risolti utilizzando blocchi condizionali, blocchi condizionali nidificati e blocchi condizionali che prevedono l'utilizzo della logica booleana.

Per la risoluzione degli esercizi si suggerisce l'utilizzo del compilatore online:

[https://www.onlinegdb.com/online\\_c++\\_compiler](https://www.onlinegdb.com/online_c++_compiler)

**ESERCIZI CON SELEZIONE SEMPLICE – IF ELSE**

1) Realizzare un programma che permetta di inserire due numeri interi e determini il numero maggiore.

2) Realizzare un programma che permetta di inserire un numero e determinare il suo valore assoluto. (Un numero negativo deve diventare positivo, un numero positivo deve restare positivo).

3) Realizzare un programma che permetta di inserire l'età di un utente e determini se l'utente è maggiorenne oppure no.

4) Realizzare un programma che permetta di inserire il valore di un carattere e determini se il valore inserito è una vocale (ATTENZIONE: utilizzare il tipo dati char).

5) Realizzare un programma che permetta di inserire i coefficienti  $a$ ,  $b$ ,  $c$  di un'equazione di secondo grado e ne calcoli, se possibile, le soluzioni.

6) Realizzare un programma che permetta di inserire due differenti orari espressi nella forma ore, minuti e secondi (CONSIGLIO: utilizzare tre differenti variabili, una per le ore, una per i minuti, ed una per i secondi) e riporti quale dei due orari viene prima.

7 **NEW**) Realizzare un programma che permetta di inserire un numero intero e che determini se il numero inserito è positivo, negativo oppure zero.

## **ESERCIZI CON SELEZIONE NIDIFICATA**

**8)** Realizzare un programma che permetta di inserire tre numeri interi e determini se questi numeri possano rappresentare le lunghezze dei lati di un triangolo. Ricorda: affinché un triangolo possa essere tale, la somma di ogni coppia lati deve essere maggiore dell'altro.

**9)** Realizzare un programma che legga tre valori rappresentanti le lunghezze dei lati di un triangolo e riporti se il triangolo è scaleno, isoscele o equilatero.

**10)** Realizzare un programma che permetta di inserire tre numeri interi e determini il valore maggiore.

**11)** Realizzare un programma che permetta di inserire tre numeri interi e determini il valore minore.

**12)** Realizzare un programma che permetta di inserire tre numeri interi (i.e., A, B, e C) e determini se B è compreso tra A e C.

**13)** Realizzare un programma che permetta di applicare gli sconti applicati da un negozio. Nello specifico, se il prezzo degli acquisti è minore di 50€ il negozio effettua lo sconto del 10%. Se il prezzo degli acquisti è maggiore di 50€ e minore di 100€, il negozio effettua lo sconto del 20%. Se il prezzo degli acquisti è maggiore di 100€, il negozio effettua lo sconto del 30%. Il programma deve chiedere all'utente di inserire il prezzo relativo ai propri acquisti e mostri a schermo lo sconto ed il prezzo finale da pagare.

Spesa Totale	Sconto Applicato
totale<50	10%
50<=totale<=100	20%
totale>100	30%

**14)** Gli abbonamenti della metropolitana di Milano possono essere settimanali (S), mensili (M), o annuali (A) e valgono per la sola zona centrale (tipo 1), per la sola zona periferica (tipo 2) o per entrambe le zone (tipo 3). I costi sono quelli riportati nella seguente tabella. Realizzare un algoritmo che calcola il costo dell'abbonamento a partire dalla durata e dal tipo.

Tipo	Settimanale	Mensile	Annuale
1	10 Euro	30 Euro	250 Euro
2	5 Euro	20 Euro	150 Euro
3	15 Euro	40 Euro	300 Euro

**15)** Realizzare un programma per determinare le tasse pagate da un lavoratore sulla base delle aliquote IRPEF 2022. Nello specifico la percentuale di tasse viene determinata in funzione del reddito percepito come riportato nella seguente tabella.

Reddito imponibile	Aliquota	Imposta dovuta
redditi fino a 15.000 euro	23%	23% sull'intero importo
da 15.001 fino a 28.000 euro	25%	3.450 euro + 25% sul reddito che supera i 15.000 euro fino a 28.000 euro

da 28.001 fino a 50.000 euro	35%	6.700 euro + 35% sul reddito che supera i 28.000 euro fino a 50.000 euro
oltre 50.001 euro	43%	14.400 euro + 43% sul reddito che supera i 50.000 euro