

# Esercitazione 3



UNIVERSITÀ  
DEGLI STUDI  
DI GENOVA

Dibris

Gualtiero Volpe  
[gualtiero.volpe@unige.it](mailto:gualtiero.volpe@unige.it)

# Soluzione Esercizi Seconda Esercitazione

Trovate le soluzioni di questi esercizi qui:

[https://github.com/gualtierovolpe/fondamenti di informatica 2023-24](https://github.com/gualtierovolpe/fondamenti_di_informatica_2023-24)

Le soluzioni delle esercitazioni vengono rilasciate il giorno prima dell'esercitazione successiva.



# Dalla Seconda Esercitazione

## Prova intermedia del 9 gennaio 2017, esercizio 4

Si scriva il valore assunto dalle variabili  $a$  e  $b$  dopo l'esecuzione delle seguenti righe di codice (*attenzione: è richiesto riportare i passaggi seguiti per ottenere la soluzione*):

```
1 int x = 2;  
  int y = 3;  
  double a = (x + y) / 2;
```

```
2 char b = 'Q'; char p = 'b';  
  b = b - 'A' + 'a';  
  b += (int(p - 'a'));
```



# Dalla Seconda Esercitazione

## Esercizio 2.1 – Esprimere una durata temporale in modi diversi

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero per esprimere una durata misurata in secondi e calcoli e stampi a video la stessa durata espressa in numero di settimane, di giorni, di ore, di minuti e di secondi.*

### Traccia per la soluzione

- Leggere la durata in secondi.
- Il numero di settimane è pari a tale durata diviso per il numero di secondi in una settimana.
- Preso il resto della divisione, il numero di giorni è pari a tale resto diviso per il numero di secondi in un giorno.
- Preso il resto della divisione, il numero di ore è pari a tale resto diviso per il numero di secondi in un'ora.
- Preso il resto della divisione, il numero di minuti è pari a tale resto diviso per il numero di secondi in un minuto.
- Il resto di quest'ultima divisione è pari al numero di secondi restanti.

# Dalla Seconda Esercitazione

## Esercizio 2.1 – Esprimere una durata temporale in modi diversi

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera un numero intero per esprimere una durata misurata in secondi e calcoli e stampi a video la stessa durata espressa in numero di settimane, di giorni, di ore, di minuti e di secondi.*

```
int main ()  
{  
    const int secondi_in_settimana = 60 * 60 * 24 * 7;  
    const int secondi_in_giorno = 60 * 60 * 24;  
    const int secondi_in_ora = 60 * 60;  
    const int secondi_in_minuto = 60;  
  
    int durata = 0;  
    int settimane = 0, ore = 0, giorni = 0, minuti = 0, secondi = 0;  
  
    std::cout << "Inserire una durata in secondi: ";  
    std::cin >> durata;  
    std::cout << std::endl;  
  
    // Calcolo del numero di settimane  
    settimane = durata / secondi_in_settimana;  
    durata = durata % secondi_in_settimana;  
  
    // Calcolo del numero di giorni  
    giorni = durata / secondi_in_giorno;  
    durata = durata % secondi_in_giorno;  
  
    // Calcolo del numero di ore  
    ore = durata / secondi_in_ora;  
    durata = durata % secondi_in_ora;  
  
    // Calcolo del numero di minuti  
    minuti = durata / secondi_in_minuto;  
    durata = durata % secondi_in_minuto;  
  
    // Calcolo del numero di secondi  
    secondi = durata;  
  
    std::cout << "La durata inserita corrisponde a: " << std::endl;  
    std::cout << settimane << " settimane" << std::endl;  
    std::cout << giorni << " giorni" << std::endl;  
    std::cout << ore << " ore" << std::endl;  
    std::cout << minuti << " minuti" << std::endl;  
    std::cout << secondi << " secondi" << std::endl;  
  
    return 0;  
}
```

# Dalla Seconda Esercitazione

E' opportuno esprimere il numero di secondi in una settimana, in un giorno, in un'ora e in un minuto come delle costanti.

```
int main ()
{
    const int secondi_in_settimana = 60 * 60 * 24 * 7;
    const int secondi_in_giorno = 60 * 60 * 24;
    const int secondi_in_ora = 60 * 60;
    const int secondi_in_minuto = 60;

    int durata = 0;
    int settimane = 0, ore = 0, giorni = 0, minuti = 0, secondi = 0;

    std::cout << "Inserire una durata in secondi: ";
    std::cin >> durata;
    std::cout << std::endl;

    // Calcolo del numero di settimane
    settimane = durata / secondi_in_settimana;
    durata = durata % secondi_in_settimana;

    // Calcolo del numero di giorni
    giorni = durata / secondi_in_giorno;
    durata = durata % secondi_in_giorno;

    // Calcolo del numero di ore
    ore = durata / secondi_in_ora;
    durata = durata % secondi_in_ora;

    // Calcolo del numero di minuti
    minuti = durata / secondi_in_minuto;
    durata = durata % secondi_in_minuto;

    // Calcolo del numero di secondi
    secondi = durata;

    std::cout << "La durata inserita corrisponde a: " << std::endl;
    std::cout << settimane << " settimane" << std::endl;
    std::cout << giorni << " giorni" << std::endl;
    std::cout << ore << " ore" << std::endl;
    std::cout << minuti << " minuti" << std::endl;
    std::cout << secondi << " secondi" << std::endl;

    return 0;
}
```

# Esercizi da Svolgere

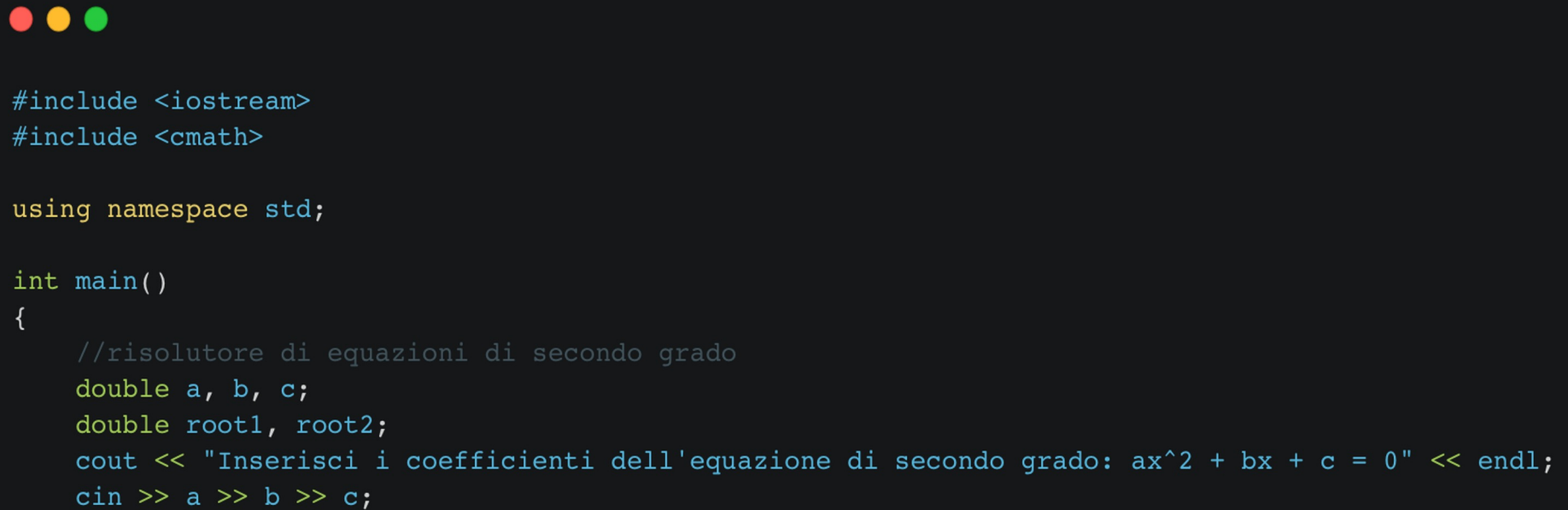
## Esercizio 3.1 – Risolutore di equazioni di secondo grado

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera i coefficienti  $a$ ,  $b$  e  $c$  (tre numeri reali) di una equazione di secondo grado nella forma  $ax^2 + bx + c = 0$  e ne calcoli le soluzioni  $x_1$  e  $x_2$ . Prima di effettuare il calcolo, il programma verificherà che l'equazione non sia degenera e che abbia soluzioni reali. In tal caso, il programma stamperà a video le soluzioni calcolate, altrimenti stamperà a video opportuni messaggi per l'utente.*

---

# Procedura per la Risoluzione

Richiedere all'utente i valori dei tre coefficienti ( $a, b, c$ )

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in C++ and is a partial implementation of a quadratic equation solver. It includes the necessary headers, uses the std namespace, and defines the main function. Inside main, it declares variables for coefficients a, b, c and roots root1, root2. It then prompts the user to input the coefficients and reads the input.

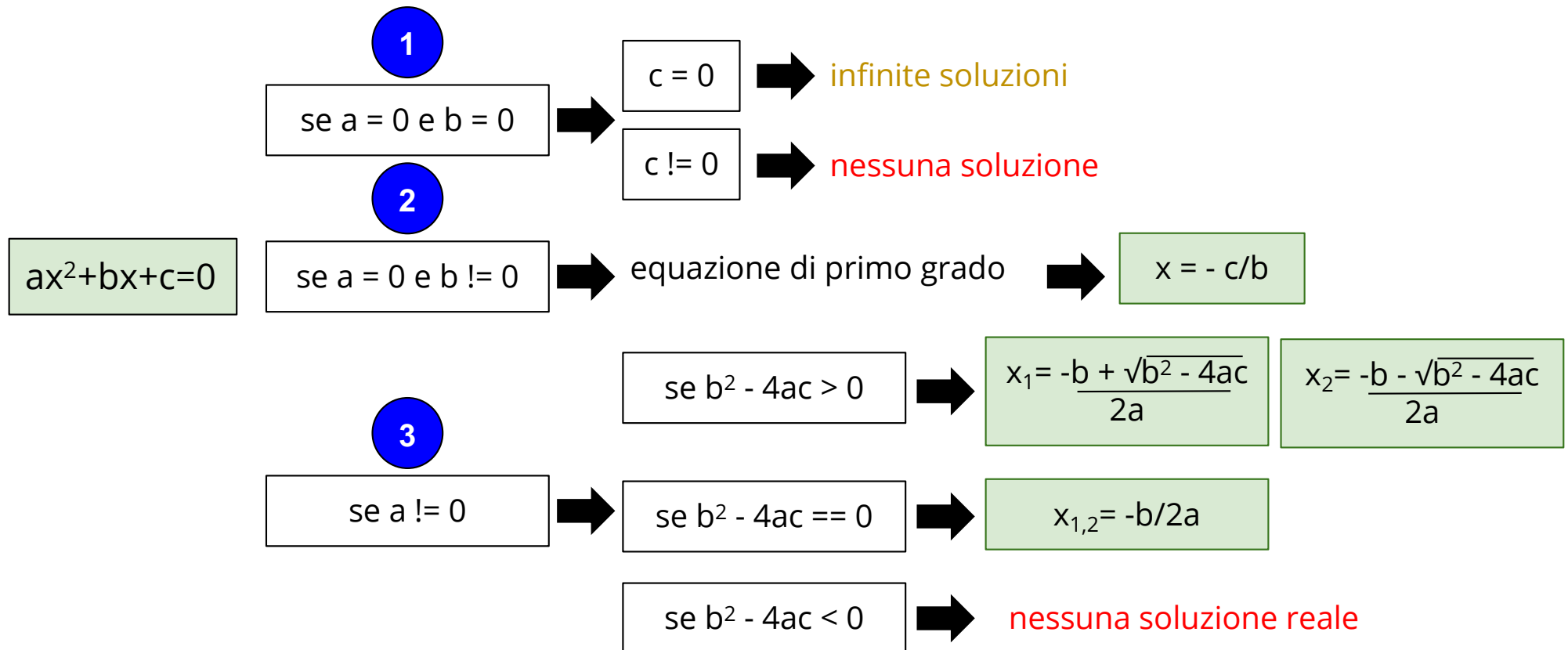
```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
    //risolutore di equazioni di secondo grado
    double a, b, c;
    double root1, root2;
    cout << "Inserisci i coefficienti dell'equazione di secondo grado: ax^2 + bx + c = 0" << endl;
    cin >> a >> b >> c;
```



# Procedura per la Risoluzione



# Procedura per la Risoluzione

Sviluppare il codice per i casi **1** e **2**

```
if (a == 0){
    if (b == 0) {
        if (c==0)
        {
            // se i tre coefficienti sono 0
            cout << "L'equazione ha infinite soluzioni" << endl;
        }
        else
        {
            // l'equazione è di tipo c = 0, con c!=0.
            cout << "L'equazione non ha soluzione" << endl;
        }
    }
    else{
        // risolviamo l'equazione di primo grado
        root1 = -c/b;
        cout << "L'equazione ha una soluzione reale:
            x = " << root1 << endl;
    }
}
```

Uno degli errori più comuni per chi inizia a programmare consiste nel confondere l'operatore assegnamento (=) con l'operatore di uguaglianza (==).

L'assegnamento (=) è utilizzato per assegnare un valore ad una variabile. L'operatore di uguaglianza (==) è utilizzato per verificare se due variabili hanno lo stesso valore.

# Procedura per la Risoluzione

Sviluppare il codice per il caso **3**

Per l'elevamento a potenza e per la radice quadrata utilizziamo le funzioni **pow** e **sqrt**.

Appartengono alla libreria standard **cmath**.

cmath è il porting della libreria math del linguaggio C.

Ogni porting di libreria dal C al C++ ha il prefisso 'c' e nessuna estensione.

Ogni elemento della libreria è definito nel namespace std.

```
else {
    if ((pow(b,2)-4*a*c)>=0){
        root1 = (-b - sqrt(pow(b,2)-4*a*c))/(2*a);
        root2 = (-b + sqrt(pow(b,2)-4*a*c))/(2*a);
        if (root1==root2){
            cout << "L'equazione ha due soluzione reali e coincidenti: x = " << root1 << endl;
        }
        else{
            cout << "L'equazione ha due soluzione reali distinte: x = " << root1 << " e x = "
                << root2 << endl;
        }
    }
    else{
        // se il discriminante è minore di zero, l'equazione non ha soluzioni reali
        cout << "L'equazione non ha soluzioni reali!" << endl;
    }
}
```

Elenco completo delle librerie standard in C++ <https://cplusplus.com/reference/>

# Esercizi da Svolgere

## Esercizio 3.2 – Verifica della validità di una password

*Scrivere un programma in linguaggio C++ che chieda all'utente di immettere da tastiera quattro caratteri e verifichi che tra essi ci sia almeno una lettera minuscola, una lettera maiuscola e una cifra. Utilizzare l'istruzione condizionale if.*

## Esercizio 3.3 – Verifica della presenza di un punto all'interno di un rettangolo

*Scrivere un programma in linguaggio C++ che chieda all'utente di immettere da tastiera le coordinate  $(x, y)$  di un punto e le coordinate  $(a, b)$  e  $(c, d)$  dei vertici rispettivamente in alto a sinistra e in basso a destra di un rettangolo e stampi a video un messaggio per indicare all'utente se il punto si trova all'interno del rettangolo.*

---

# Esercizi da Svolgere

## Esercizio 3.4 – Calcolo dei giorni di cui si compone un mese specificato

*Scrivere un programma in linguaggio C++ che chieda all'utente di immettere da tastiera mese e anno (due numeri interi) e calcoli e stampi a video il numero di giorni di cui si compone il mese specificato. Per il mese di febbraio, si ricorda che un anno è bisestile se il suo numero è divisibile per 4, con l'eccezione che gli anni secolari (quelli divisibili per 100) sono bisestili solo se divisibili per 400.*

---

# Esercizi da Svolgere

## Esercizio 3.5 – Prova intermedia dell'8 gennaio 2016, Esercizio 1

*Si scriva un programma C++ che operi come segue:*

- 1. Chieda all'utente di inserire da tastiera giorno e mese (due numeri interi).*
- 2. Verifichi che i valori inseriti dall'utente siano validi, ovvero che il mese sia compreso tra 1 e 12 e, per semplicità, che il giorno sia compreso tra 1 e 31. Nel caso in cui i valori non siano validi, il programma stamperà a video un messaggio di errore e terminerà con valore di ritorno -1.*
- 3. Stampi a video la stagione corrispondente alla data immessa:*
  - a. "Primavera" se la data è compresa tra il 21 marzo e il 20 giugno;*
  - b. "Estate" se la data è compresa tra il 21 giugno e il 22 settembre;*
  - c. "Autunno" se la data è compresa tra il 23 settembre e il 21 dicembre;*
  - d. "Inverno" se la data è compresa tra il 22 dicembre e il 20 marzo.*

# Istruzioni Finali

Cercate di svolgere gli esercizi in autonomia/a piccoli gruppi  
(è il miglior modo per vedere se avete capito)

Per ogni domanda, siamo a disposizione

Troverete le soluzioni di questi esercizi qui:

[https://github.com/gualtierovolpe/fondamenti\\_di\\_informatica\\_2023-24](https://github.com/gualtierovolpe/fondamenti_di_informatica_2023-24)

Le soluzioni delle esercitazioni vengono rilasciate il giorno prima dell'esercitazione successiva.

---