

# Esercitazione 2



UNIVERSITÀ  
DEGLI STUDI  
DI GENOVA

| Dibris

Gualtiero Volpe  
gualtiero.volpe@unige.it

# Soluzione esercizi prima esercitazione

Trovate le soluzioni di questi esercizi qui:

[https://github.com/gualtierovolpe/fondamenti\\_di\\_informatica\\_2025-26](https://github.com/gualtierovolpe/fondamenti_di_informatica_2025-26)

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

# Un po' di teoria

## **Prova intermedia del 31 gennaio 2018, esercizio 4**

Scrivere quale componente di un calcolatore è responsabile di ciascuna delle seguenti operazioni:

1. Contiene le istruzioni e i dati dei programmi in esecuzione: \_\_\_\_\_
2. Controlla l'attività dell'intero calcolatore: \_\_\_\_\_
3. Trasferisce dati da un componente all'altro: \_\_\_\_\_
4. Salva i dati in modo permanente: \_\_\_\_\_

# Un po' di teoria

## Prova intermedia del 31 gennaio 2018, esercizio 2

Per ciascuno degli intervalli sotto elencati, si completino le espressioni C++ corrispondenti, indicando quali operatori relazionali e logici debbano essere applicati affinché la variabile intera  $y$ , risultato della valutazione dell'espressione, assuma il valore *Vero* se e solo se il valore corrente della variabile intera  $x$  appartiene all'intervallo.

- 1     $x \in [6, 10]$                      $y = (x \text{ \_\_\_\_\_\_ } 6) \text{ \_\_\_\_\_\_ } (x \text{ \_\_\_\_\_\_ } 10);$
- 2     $x \in (-\infty, 0]$                     $y = (x == 0) \text{ \_\_\_\_\_\_ } (x \text{ \_\_\_\_\_\_ } 0);$
- 3     $x \in (2, 4) \cup (5, 8]$             $y = ((x > 2) \text{ \_\_\_\_\_\_ } (x \text{ \_\_\_\_\_\_ } 4)) \text{ \_\_\_\_\_\_ } ((x \text{ \_\_\_\_\_\_ } 5) \text{ \_\_\_\_\_\_ } (x \leq 8));$
- 4     $x \in [-3, 3], x \neq 0$             $y = (x \text{ \_\_\_\_\_\_ } -3) \text{ \_\_\_\_\_\_ } (x \text{ \_\_\_\_\_\_ } 3) \text{ \_\_\_\_\_\_ } (x \text{ \_\_\_\_\_\_ } 0);$

# Un po' di teoria

## 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');
```

# Esercizi da Svolgere

## Esercizio 2.1 – **Risolutore di equazioni di primo grado**

*Si scriva un programma in linguaggio C++ che chieda all'utente di immettere da tastiera i coefficienti  $a$  e  $b$  (due numeri reali) di un'equazione di primo grado nella forma  $ax + b = 0$ , ne calcoli la soluzione e la stampi a video. Prima di effettuare il calcolo, il programma verificherà che la soluzione esista e che l'equazione non abbia infinite soluzioni. In tali casi, il programma stamperà a video un messaggio appropriato.*

# Esercizio 2.1: Procedura per risolvere l'esercizio

1. Dichiarare due variabili di tipo reale per i valori dei coefficienti  $a$  e  $b$ .
2. Utilizzare gli oggetti di I/O cin e cout per ottenere i due numeri tramite il terminale, salvando il contenuto nelle variabili corrispondenti.
3. Se l'equazione è degenere (cioè  $a$  vale zero):  
Se anche  $b$  vale zero, l'equazione ha infinite soluzioni, altrimenti non esiste soluzione.  
In questi casi, stampare a video un messaggio appropriato.
4. Altrimenti (equazione non degenere), calcolare la soluzione  $x = -b/a$ .
5. Stampare il risultato sul terminale.

# Esercizio 2.1: Soluzione

```
src > esercizio_02_01 > main.cpp > main()
1  #include <iostream>
2
3  int main() {
4
5      float a = 0.0f; float b = 0.0f;
6      float x = 0.0f;
7
8      // Acquisizione coefficienti
9      std::cout << "Inserire i coefficienti a e b di una equazione di primo grado" << std::endl;
10     std::cout << "Coefficiente a: ";
11     std::cin >> a;
12     std::cout << "Coefficiente b: ";
13     std::cin >> b;
14
15     // Caso degenere
16     if (a == 0)
17     {
18         if (b == 0)
19             std::cout << "L'equazione ha infinite soluzioni" << std::endl;
20         else
21             std::cout << "L'equazione non ha soluzione" << std::endl;
22     }
23     // Caso non degenere
24     else {
25         x = - b / a;
26         std::cout << "La soluzione vale x = " << x << std::endl;
27     }
28     return 0;
29 }
30
```




# Esercizi da Svolgere

## Esercizio 2.2 – **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 un'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 degenere 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 ( $a$ ,  $b$ ,  $c$ )

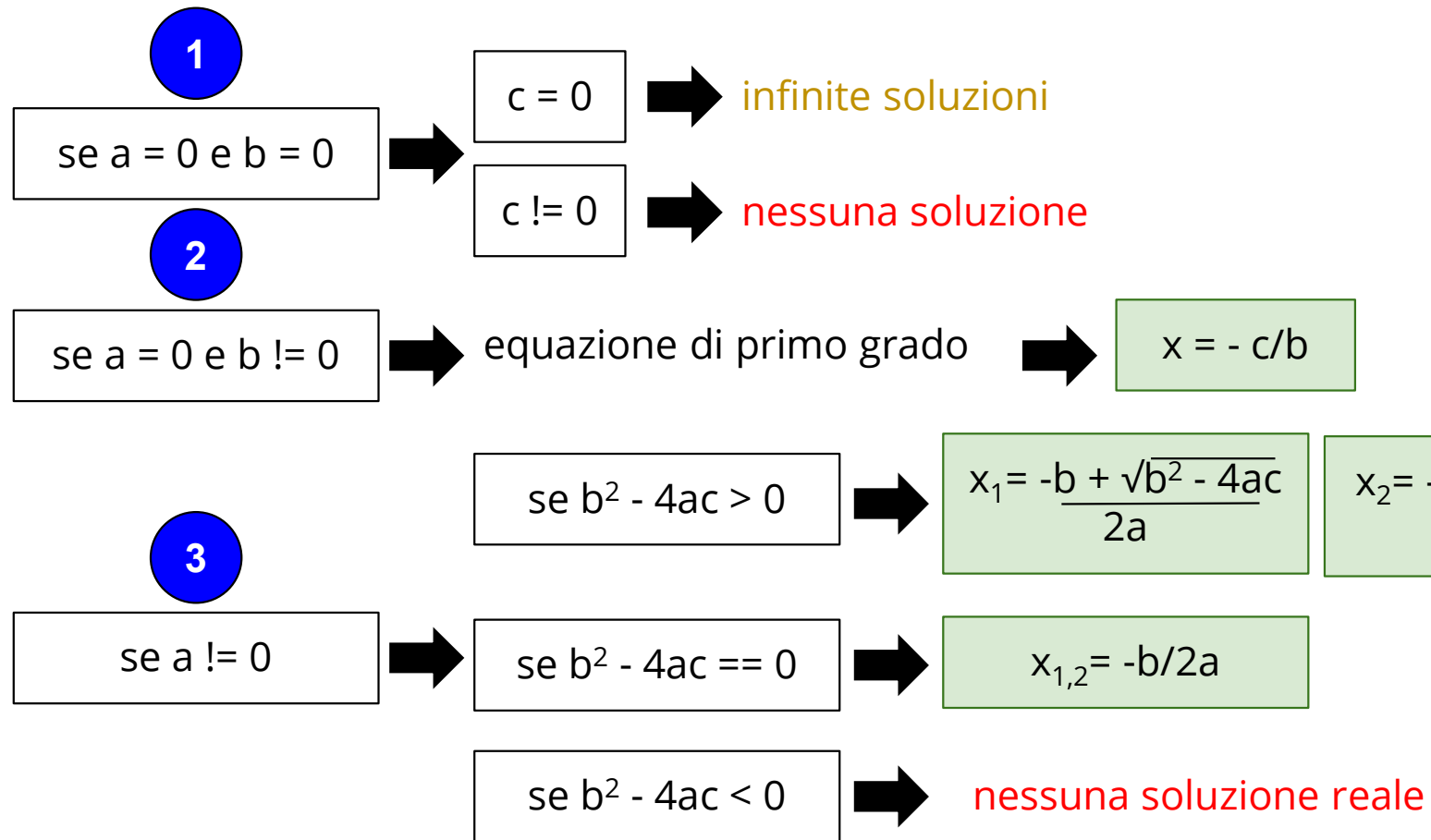


```
#include <iostream>
#include <cmath>

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

```
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;  
    }  
}
```

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.

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

# Esercizi da Svolgere

## Esercizio 2.3 – 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 2.4 – 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 2.5 – **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 2.6 – 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\\_2025-26](https://github.com/gualtierovolpe/fondamenti_di_informatica_2025-26)

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