

Linguaggio cpp / c

Struttura di un programma in linguaggio C

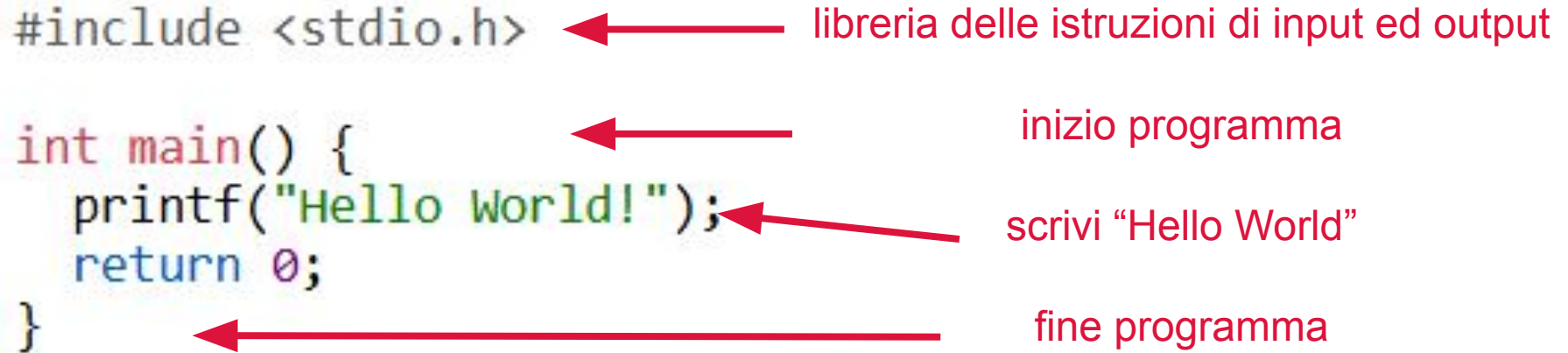
```
#include <stdio.h>
int main() {
    printf("Hello World!");
    return 0;
}
```

libreria delle istruzioni di input ed output

inizio programma

scrivi "Hello World"

fine programma

The diagram illustrates the structure of a C program. It shows a code snippet with four lines: a preprocessor directive, a function definition, a function call, and a return statement. Red arrows point from descriptive text labels to specific parts of the code. The first arrow points from 'libreria delle istruzioni di input ed output' to the include directive. The second arrow points from 'inizio programma' to the opening brace of the main function. The third arrow points from 'scrivi "Hello World"' to the printf statement. The fourth arrow points from 'fine programma' to the closing brace of the main function.

Struttura di un programma in linguaggio C++

```
#include <iostream>  
using namespace std;
```

libreria delle istruzioni di input ed output



```
int main() {  
    cout << "Hello World!";  
    return 0;  
}
```

inizio programma



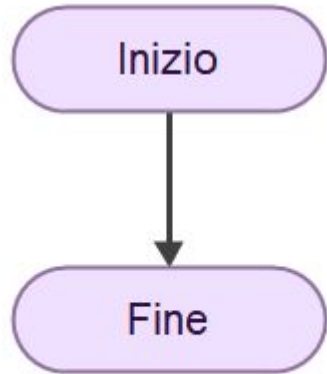
scrivi "Hello World"



fine programma



Flowgorithm



linguaggio C

```
#include <stdio.h>
```

```
int main() {
```

```
    return 0;
```

```
}
```

e linguaggio C++

```
#include <iostream>  
using namespace std;
```

```
int main() {
```

```
    return 0;
```

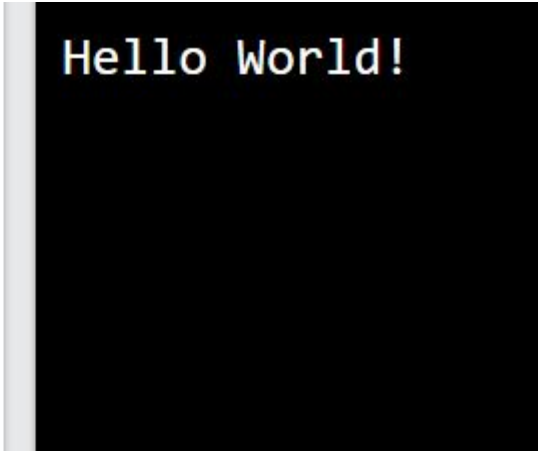
```
}
```

Il primo programma

Registratevi su [onlinegdb](#) e scrivete il primo programma

```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```

A screenshot of a terminal window with a black background. The text "Hello World!" is displayed in a white, monospaced font at the top of the window.

Hello World!

Il primo programma

```
#include <stdio.h>
```

```
int main() {  
    printf("Hello World!");  
    return 0;  
}
```

```
#include <iostream>  
using namespace std;
```

```
int main() {  
    cout << "Hello World!";  
    return 0;  
}
```



Hello World!

Uso dei caratteri di “escape” \t e \n

Carattere	Nome	Funzione	Risultato Visivo
<code>\n</code>	Newline	Va a capo	Interruzione di riga
<code>\t</code>	Tab	Spazio ampio	Inserisce uno spazio lungo (indentazione)

linguaggioC

```
1  #include <stdio.h>
2  int main()
3  {
4      printf("LISTA SPESA:\n- Pane");
5  }
6
7
8
9
10
11
```

STDIN

Input for the program (Optional)

Output:

LISTA SPESA:

- Pane

linguaggio C++

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "LISTA SPESA:\n- Pane";
7
8      return 0;
9  }
```

STDIN

Input for the program (Optional)

Output:

LISTA SPESA:
- Pane

Esempio “C”

Output:

LISTA SPESA:

- Pane
- Latte
- Caffè

```
#include <stdio.h>
int main(){
    printf("LISTA SPESA:\n\t- Pane\n\t- Latte\n\t- Caffè");
}
```

Esempio “C++”

Output:

LISTA SPESA:

- Pane
- Latte
- Caffè

```
#include <iostream>
using namespace std;

int main()
{
    cout << "LISTA SPESA:\n\t- Pane\n\t- Latte\n\t- Caffè";

    return 0;
}
```

Programma: risultati **Calcio serie A**

Home	Visitors	Score

Juventus	Genoa	2-0
Napoli	Milan	1-0
Inter	Roma	5-2

Uso del carattere tabulatore

```
#include <iostream>
using namespace std;

int main() {
    cout << "Home    \tVisitors \t Score"<< endl;
    cout << "-----"<< endl;
    cout << "Juventus \tGenoa   \t 2-0"<< endl;
    cout << "Napoli   \tMilan   \t 1-0"<< endl;
    cout << "Inter    \tRoma    \t 5-2"<< endl;
    cout << "-----"<< endl;

    return 0;
}
```

Programma

```
#include <iostream>
using namespace std;

int main() {
    cout << "Home    \tVisitors \t Score"<< endl;
    cout << "-----"<< endl;
    cout << "Juventus \tGenoa    \t 2-0"<< endl;
    cout << "Napoli   \tMilan    \t 1-0"<< endl;
    cout << "Inter    \tRoma     \t 5-2"<< endl;
    cout << "-----"<< endl;

    return 0;
}
```

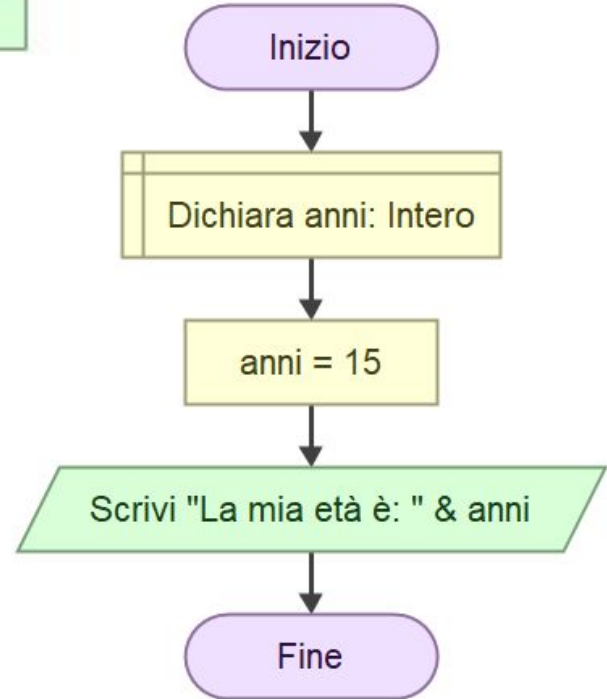
Le variabili

La variabile intera

La mia età è: 15

```
#include <iostream>
using namespace std;
```

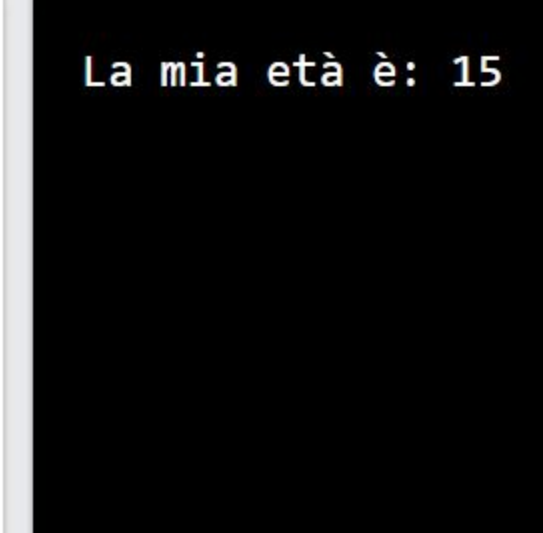
```
int main() {
    int anni;
    anni = 15;
    cout << "La mia età è: " << anni;
    return 0;
}
```



Risultato

```
#include <iostream>
using namespace std;

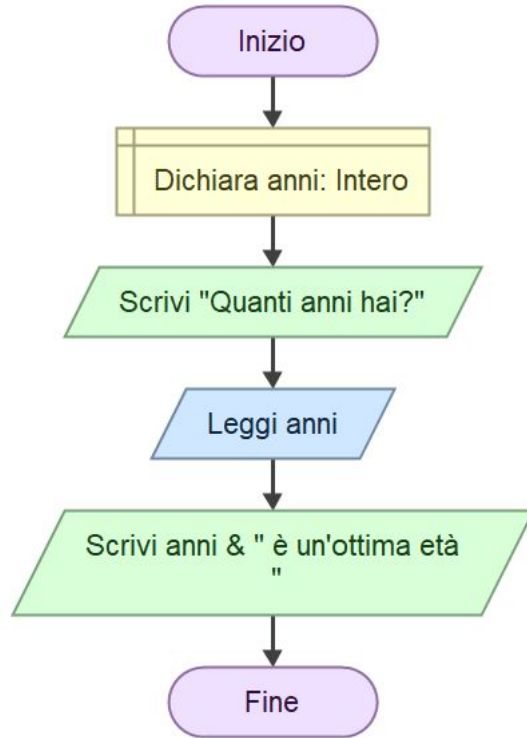
int main() {
    int anni;
    anni = 15;
    cout << "La mia età è: "<<anni;
    return 0;
}
```



La mia età è: 15

ESERCIZIO: aggiungi la parola "anni" alla fine del rigo

Input di un dato in C++

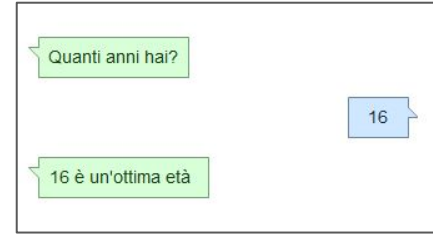


```
#include <iostream>
using namespace std;
```

```
int main() {
    int anni;
```

```
    cout << "Quanti anni hai ? ";
    cin >> anni;
    cout << anni << " è un'ottima età ";
```

```
    return 0;
}
```



main.cpp



Share

Run

Output

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int anni;
6
7     cout << "Quanti anni hai ? ";
8     cin >> anni;
9     cout << anni << " è un'ottima età ";
10
11     return 0;
12 }
13
```

Quanti anni hai ? 55

55 è un'ottima età

=== Code Execution Successful ===

Programma: somma di due numeri

```
#include <iostream>
using namespace std;

int main() {
    int x, y;
    int somma;
    cout << "Inserisci un numero: ";
    cin >> x;
    cout << "Inserisci un altro numero: ";
    cin >> y;
    somma = x + y;
    cout << "La somma è: " << somma;
    return 0;
}
```

Output

main.cpp



Share

Run

Output

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x, y;
6      int somma;
7      cout << "Inserisci un numero: ";
8      cin >> x;
9      cout << "Inserisci un altro numero: ";
10     cin >> y;
11     somma = x + y;
12     cout << "La somma è: " << somma;
13     return 0;
14 }
15
```

Inserisci un numero:

71

Inserisci un altro numero: 45

La somma è: 116

=== Code Execution Successful ===

Variabili C++

Le variabili sono contenitori per l'archiviazione di valori di dati.

In C++ esistono diversi tipi di variabili (definite con parole chiave diverse), ad esempio:

- **int**- memorizza numeri **interi** (numeri interi), senza decimali, come 123 o -123
- **double**- memorizza numeri in **virgola mobile**, con decimali, come 19,99 o -19,99
- **char**- memorizza singoli **caratteri**, come 'a' o 'B'. I valori dei caratteri sono racchiusi tra virgolette singole.
- **string**- memorizza **testo**, come "Hello World". I valori stringa sono racchiusi tra virgolette doppie.
- **bool**- memorizza valori con due stati: **vero o falso**

Output delle variabili in C

- Per gestire l'output delle variabili in C si usa principalmente la funzione `printf` (che sta per *print formatted*).
- A differenza di altri linguaggi, in C devi specificare esplicitamente al computer **che tipo di dato** stai cercando di stampare usando i cosiddetti **specificatori di formato**.

```
printf("Int: %d\n", a);  
printf("Float (7 cifre): %.7f\n", b);  
printf("Double (10 cifre): %.10lf\n", c);
```

Linguaggio C

```
1  #include <stdio.h>
2
3  int main() {
4      int a = 10.9;
5      float b = 1.123456789f;
6      double c = 1.123456789;
7
8      printf("Int: %d\n", a);
9      printf("Float (7 cifre): %.7f\n", b);
10     printf("Double (10 cifre): %.10lf\n", c);
11 }
12
13
```

STDIN

Input for the program (Optional)

Output:

Int: 10

Float (7 cifre): 1.1234568

Double (10 cifre): 1.1234567890

Linguaggio C++

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int a = 10.9;      // Diventa 10 (perde i decimali)
6     float b = 10.123456789f; // Arrotonda dopo circa 7 cifre
7     double c = 10.123456789; // Mantiene la precisione alta
8
9     cout << "Int: " << a << endl;
10    cout << "Float: " << b << endl;
11    cout << "Double: " << c;
12 }
13
```

STDIN

Input for the program (Optional)

Output:

Int: 10

Float: 10.1235

Double: 10.1235

Dichiarare più variabili

```
#include <iostream>
using namespace std;

int main() {
    int x = 5, y = 6, z = 50;
    cout << x + y + z;
    return 0;
}
```

Esercizi di Output (Escape Characters)

1. **La Tabella:** Stampa tre nomi (es. Marco, Anna, Luca) incolonnati sotto l'intestazione "NOME", usando `\t` per allinearli. Aggiungi la colonna "CITTA"
2. **L'Elenco a Capo:** Stampa la frase "Pane, Latte, Uova" facendo in modo che ogni parola finisca su una riga diversa usando solo un `printf` (o `cout`) e il carattere `\n`.
3. **Il Triangolo:** Usa gli spazi e `\n` per stampare un piccolo triangolo fatto di asterischi (*) alto 3 righe con un solo `printf` / `cout`.
4. **Citazione Testuale:** Prova a stampare la frase: `L'uso del "double" è importante.` (Suggerimento: per le virgolette interne serve l'escape `\`).

Esercizi sui Tipi di Dato (Precisione e Perdita)

Il Ladro di Decimali: Crea una variabile `double` con valore `15.99`. Assegnala a una variabile `int` e stampa tutto. Cosa è successo al `.99`?

Divisione tra Interi: Crea due variabili `int a = 10` e `int b = 3`. Calcola `a / b` e stampa il risultato. Perché non esce `3.33`?

Il Limite del Float: Crea un `float` con 10 cifre decimali (es. `1.123456789`). Stampalo con `%.10f` in C o `setprecision(10)` in C++. Osserva dopo quale cifra il numero inizia a diventare impreciso.

Esercizi sui Tipi di Dato (Precisione e Perdita)

Media Voti: Chiedi all'utente tre voti (usa `int`). Calcola la media e salvala in un `double`. Stampa la media con due cifre decimali (es. `%.2f`).

Prezzi e IVA: Crea una variabile `double` per il prezzo di un prodotto (es. `25.50`). Calcola l'IVA al 22% e stampa il prezzo finale. Usa il `double` per evitare errori di arrotondamento.

Confronto Diretto: Scrivi un programma che memorizza lo stesso numero decimale lungo (es. `2.7182818284`) in un `float` e in un `double`. Stampali entrambi con 10 decimali e commenta la differenza.

Soluzioni (C)

1. La Tabella (\t)

C



```
#include <stdio.h>
int main() {
    printf("NOME\tCITTA\n");
    printf("Marco\tRoma\nAnna\tMilano\nLuca\tNapoli\n");
}
```

2. L'Elenco a Capo (\n)

C



```
#include <stdio.h>
int main() {
    printf("Pane\nLatte\nUova\n");
}
```

Soluzioni (C)

3. Il Triangolo

C



```
#include <stdio.h>
int main() {
    printf(" * \n *** \n*****\n");
}
```

4. Citazione con Escape (\")

C



```
#include <stdio.h>
int main() {
    printf("L'uso del \"double\" e' importante.\n");
}
```

Soluzioni (C)

5. Il Ladro di Decimali (Cast Implicito)

C



```
#include <stdio.h>
int main() {
    double prezzo = 15.99;
    int intero = prezzo; // Il compilatore taglia i decimali
    printf("Double originale: %.2f\nIntero risultante: %d\n", prezzo, intero);
}
```

6. Divisione tra Interi

C



```
#include <stdio.h>
int main() {
    int a = 10, b = 3;
    printf("Risultato intero: %d\n", a / b); // Risultato: 3
    printf("Risultato reale: %.2f\n", (double)a / b); // Risultato: 3.33
}
```

Soluzioni (C)

7. Il Limite del Float

C



```
#include <stdio.h>
int main() {
    float b = 1.123456789f;
    printf("Float a 10 decimali: %.10f\n", b); // Noterai errori dopo la 7a cifra
}
```

8. Media Voti

C



```
#include <stdio.h>
int main() {
    int v1 = 28, v2 = 25, v3 = 30;
    double media = (v1 + v2 + v3) / 3.0; // Usare 3.0 forza il calcolo in double
    printf("Media: %.2lf\n", media);
}
```

Soluzioni (C)

9. Prezzi e IVA

C



```
#include <stdio.h>
int main() {
    double prezzo = 25.50;
    double prezzo_iva = prezzo * 1.22;
    printf("Prezzo finale: %.2f Euro\n", prezzo_iva);
}
```

10. Confronto Diretto

C



```
#include <stdio.h>
int main() {
    float f = 2.7182818284f;
    double d = 2.7182818284;
    printf("Float: %.10f\n", f);
    printf("Double: %.10lf\n", d); // Il double sara' molto piu' fedele al numero
}
```