
Ingresso e Uscita Variabili e costanti intere

Operatore di uscita <<

- Scrittura formattata su *cout*
- *cout<<obj1<<obj2<<...<<endl ;*
- Il generico oggetto da stampare può anche essere una *variabile* o una *costante* con nome

Riepilogo iostream

- Ogni file sorgente che contenga riferimenti ad oggetti della libreria di ingresso/uscita deve contenere le direttive

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

- Devono precedere il primo punto in cui viene utilizzato uno stream di ingresso/uscita

Riepilogo primo esercizio

```
#include <iostream>  
  
using namespace std;  
  
main()  
  
{  
  
    cout<<"Ciao mondo!\n" ;  
  
}
```

Riepilogo secondo esercizio

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

```
    cout<<"Ciao mondo!"<<endl ;
```

```
}
```

Riepilogo compilazione

- Sintassi più semplice per generare un programma eseguibile da un file sorgente:
- *g++ nome_sorgente.cc*
 - Assegna un nome predefinito al programma eseguibile (a.out)
- *g++ -o nome_eseguibile nome_sorgente.cc*
 - Permette di scegliere il nome del programma eseguibile

Esercizio 1/2

- Scrivere un programma in cui si definisce una variabile intera e se ne stampa il valore sullo schermo

Esercizio 2/2

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    int i = 10 ;  
  
    cout<<i;  
  
}
```

Esercizio 1/2

- Scrivere un programma in cui si definisce una variabile intera e se ne stampa il valore sullo schermo col seguente formato:

Il valore della variabile è 10.

- E si va a capo

Esercizio 2/2

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
int i = 10 ;  
  
cout<<"Il valore della variabile è  
"<<i<<". "<<endl ;  
  
}
```

Una soluzione alternativa

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
int i = 10 ;  
  
cout<<"Il valore della variabile è "<<i ;  
  
cout<<". "<<endl ;  
  
}
```

Ritorno valore finale

- Un processo ritorna un valore quando termina
- Può essere letto ed utilizzato dal programma che lo ha invocato
- La funzione *main* può (in effetti dovrebbe sempre) essere dichiarata ritornare un valore di tipo intero
- Il valore di ritorno della funzione *main* è il valore ritornato dal processo stesso

Esempio

```
#include <iostream>  
using namespace std;  
  
int main()  
  
{  
  
int i = 10 ;  
  
cout<<"Il valore della variabile è "<<i ;  
  
cout<<". "<<endl;  
  
return 0 ;  
  
}
```

Esercizio 1/2

- Scrivere un programma in cui si definisce una variabile intera e se ne stampa il valore sullo schermo, quindi se ne cambia il valore e si stampa di nuovo il (nuovo) valore sullo schermo

Esercizio 2/2

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

main()
{
    int i = 10 ;

    cout<<"Il valore della variabile è "<<i<<endl ;

    i = 12 ;

    cout<<"Il nuovo valore è "<<i<<endl ;

}
```

Esercizio 1/2

- Scrivere un programma in cui si definisce una costante intera e se ne stampa il valore sullo schermo col seguente formato:

Il valore della costante è 10.

- E si va a capo

Esercizio 2/2

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    const int i = 10 ;  
  
    cout<<"Il valore è "<<i<<". "<<endl ;  
  
}
```

Lettura dallo *stdin*

- Operatore di ingresso `>>` applicato ad un oggetto di tipo *istream*
- *Esempio: cin>>nome_variabile ;*
- Legge i caratteri in ingresso dallo standard input
- Li interpreta in base al tipo della variabile
- Assegna il valore letto alla variabile di nome *nome_variabile*

Esempio di interpretazione

- `cin >> a;`
- La variabile `a` è di tipo `int`
- Si leggono i caratteri `2`, `3` e `\n`
- Vengono interpretato come le due cifre decimali del numero `23`
- Il numero `23` viene memorizzato nella variabile `a`

Esercizio 1/2

- Si scriva un programma che legge un valore intero da tastiera e lo stampa a video

Esercizio 2/2

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
    int i ;  
  
    cin>>i ;  
  
    cout<<"Il valore inserito è "<<i<<endl ;  
  
}
```

Ingresso inconsistente

- Cosa accade se la sequenza di caratteri letta non rappresenta alcun numero in notazione decimale?
- Il valore del secondo argomento rimane invariato (non avviene alcuna memorizzazione)
- Le successive letture falliranno
- Vedremo in futuro come resettare lo stato dello stream per non fare più fallire le successive letture

Esercizio 1/2

- Miglioriamo l'esercizio precedente
- Vogliamo stampare anche un messaggio di richiesta del numero da inserire:

Inserisci un valore intero: 13

Il valore inserito è: 13

- L'operatore di ingresso `>>` applicato al *cin* **non scrive sullo standard output**

Esercizio 2/2

```
#include <iostream>  
using namespace std;  
  
main()  
  
{  
  
int i ;  
  
cout<<"Inserisci un valore intero " ;  
  
cin>>i ;  
  
cout<<"Il valore inserito è "<<i<<endl ;  
  
}
```

Esercizio 1/3

- Scrivere un programma che legge in ingresso due valori interi e stampa il risultato della moltiplicazione tra i due numeri

Inserisci il primo numero: 10

Inserisci il secondo numero: 20

*10 * 20 = 200*

Esercizio 2/3

```
#include <iostream>
using namespace std;
main()
{
    int i, j, ris ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    ris = i * j ;
    cout<<i<<"*"<<j<<" = "<<ris<<endl ;
}
```

Esercizio 3/3

```
#include <iostream> /* Soluzione alternativa: */
using namespace std; /* senza variabile di */
main() /* appoggio */
{
    int i, j ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    cout<<i<<"*"<<j<<" = "<<i*j<<endl ;
}
```

Esercizio 1/3

- Scrivere un programma che legge in ingresso due valori interi e stampa sia il risultato della divisione intera tra i due numeri che il resto della divisione stessa

Inserisci il primo numero: 5

Inserisci il secondo numero: 2

5 / 2 = 2 con resto 1

Esercizio 2/3

```
#include <iostream>
using namespace std;
main()
{
    int i, j, div, resto ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    div = i / j ;
    resto = i % j ;
    cout<<i<<" / "<<j<<" = "<<div<<" con resto
        "<<resto<<endl ;
}
```

Esercizio 3/3

```
#include <iostream>      /* Soluzione alternativa */
using namespace std;    /* senza variabili di */
main()                  /* appoggio */
{
    int i, j ;

    cout<<"Inserisci il primo numero " ;
    cin>>i ;
    cout<<"Inserisci il secondo numero " ;
    cin>>j ;

    cout<<i<<" / "<<j<<" = "<<i/j<<" con resto
        "<<i%j<<endl ;
}
```

Esercizio 1/4

- Scrivere un programma che legge in ingresso due valori interi e li memorizza in due variabili, quindi scambia il contenuto delle variabili e lo stampa sullo schermo

Inserisci il valore di i: 2

Inserisci il valore di j: 3

Dopo lo scambio: $i = 3, j = 2$

Esercizio 2/4

- Primo esercizio un po' più difficile
- Iniziamo ad acquisire la mentalità giusta
- Pensiamo prima all'algoritmo
- Forse ci serve una variabile d'appoggio ...

Esercizio 3/4

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

```
int main()
```

```
{
```

```
    int i, j ;
```

```
    cout<<"Inserisci il valore di i " ;
```

```
    cin>>i ;
```

```
    cout<<"Inserisci il valore di j " ;
```

```
    cin>>j ;
```

Esercizio 4/4

```
int appoggio = i ;
```

```
i = j ;
```

```
j = appoggio ;
```

```
cout<<"Dopo lo scambio: i = "<<i
```

```
<<" , j = "<<j ;
```

```
return 0 ;
```

```
}
```

- Scrivere un programma che legge in ingresso due valori interi e li memorizza in due variabili, quindi scambia il contenuto delle variabili e lo stampa sullo schermo
- Ma senza utilizzare **nessuna** variabile d'appoggio!
- *scambia_senza_appoggio.cc*

Altri dettagli sui processi

- Ad ogni processo è associato un identificatore numerico: *pid*
- Per elencare i propri processi:

ps x

- Per elencare tutti i processi:

ps ax

Ancora sui segnali

- **Ctrl + C** manda il segnale SIGINT (interruzione) al processo in esecuzione
- A volte tale segnale può non bastare per interrompere il processo
- Comando *kill*: spedisce segnali ai processi
kill -signal pid
- Invocato da un altro terminale

Compiti per casa 1

- Scrivere i seguenti programmi
 - **senza** utilizzare istruzioni di **controllo di flusso** (niente istruzioni condizionali ed iterative, ma solo esecuzione sequenziale)
 - facendo uso **solo** di variabili di tipo *int* e dei relativi operatori
 - $+$, $-$, $*$, $/$, $\%$, *abs()*

Compiti per casa 2

- Scrivere un programma che legge in ingresso un numero intero, lo interpreta come un tempo espresso in secondi, e lo stampa in minuti e secondi (*da_sec_a_min_sec.cc*)

Tempo in secondi? 67

Equivalgono a 1 min, 7 sec

Compiti per casa 3

- Scrivere un programma che legge in ingresso due numeri, li interpreta come un tempo espresso in minuti e secondi, e lo stampa in secondi (*da_min_sec_a_sec.cc*)
 - **Attenzione:** per semplicità assumiamo come valido anche un ingresso in cui il secondo numero sia maggiore di 59

Minuti ? 3

Secondi ? 78

Equivalgono a 258 secondi

Compiti per casa 4

- Scrivere un programma che legge in ingresso quattro numeri, li interpreta come due tempi espressi in minuti e secondi, e stampa la differenza tra i due tempi, espressa in secondi (*soluzione non fornita*)
 - **Attenzione:** per semplicità assumiamo come valido anche un ingresso in cui il secondo numero sia maggiore di 59

Minuti e secondo primo tempo ? 3 45

Minuti e secondi secondo tempo ? 5 36

Differenza: 111

Compiti per casa 5

- Scrivere un programma che legge in ingresso quattro numeri, li interpreta come due tempi espressi in minuti e secondi, e stampa la differenza tra i due tempi, di nuovo espressa in minuti e secondi (*soluzione non fornita*)
- **Attenzione:** per semplicità assumiamo come valido anche un ingresso in cui il secondo numero sia maggiore di 59

Minuti e secondo primo tempo ? 3 45

Minuti e secondi secondo tempo ? 5 36

Differenza: 1 51

Compiti per casa 6

- Scrivere un programma che legge in ingresso un numero intero e stampa 0 se il numero è pari, 1 altrimenti (*0_se_pari.cc*)

Inserisci un numero intero: 23

1

- Scrivere un programma che legge in ingresso un numero intero e stampa 1 se il numero è pari, 0 altrimenti (*1_se_pari.cc*)

Compiti per casa 7

- Scrivere un programma che legge in ingresso due numeri interi positivi, poi stampa 0 se il primo è multiplo dell'altro, 1 altrimenti (*0_se_multiplo.cc*)

Inserisci il primo numero intero positivo: 32

Inserisci il secondo numero intero positivo: 11

1

Compiti per casa 8

- Scrivere un programma che legge in ingresso due numeri interi positivi, poi stampa 1 se il primo è multiplo dell'altro, 0 altrimenti (*1_se_multiplo.cc*)

Inserisci il primo numero intero positivo: 32

Inserisci il secondo numero intero positivo: 11

0

Compiti per casa 9

- Scrivere un programma che legge in ingresso un numero intero diverso da 0, e stampa -1 se è negativo, 1 se è positivo (*1_se_pos-1_se_neg.cc*)

Inserisci un numero intero: -3

-1

Compiti per casa 10

- Scrivere un programma che legge in ingresso un numero intero diverso da 0, e stampa 0 se è negativo, 1 se è positivo (*0_se_neg_1_se_pos.cc*)

Inserisci un numero intero: -3

0

Compiti per casa 11

- Scrivere un programma che legge in ingresso un numero intero diverso da 0, e stampa 1 se è negativo, 0 se è positivo

Inserisci un numero intero: -3

1