

Fondamenti di Informatica – Input/output standard in C e in C++

Prof. PIER LUCA MONTESSORO - Ing. DAVIDE PIERATTONI
Facoltà di Ingegneria Università degli Studi di Udine

INPUT/OUTPUT STANDARD (tastiera / monitor)

In C, l'header file necessario per effettuare l'input da tastiera e l'output sul monitor è **stdio.h**:

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

In C++ le stesse operazioni richiedono l'header file **iostream**:

```
#include <iostream>
```

Lettura di dati da tastiera

Per leggere da tastiera un intero *i*, un numero reale *z* oppure una stringa *str*, il C offre la funzione **scanf** (di input formattato):

```
scanf ("%d %f %s", &i, &z, str);
```

Il C++ permette simili operazioni in modo più semplice; l'istruzione **cin** interpreta automaticamente il tipo del dato a partire da quello della variabile in cui il dato viene scritto:

```
cin >> i >> z >> str;
```

Stampa sul monitor

Per stampare sul monitor (standard output) un intero *i*, un numero reale *z* oppure una stringa *str*, in C si usa la funzione **printf** (di output formattato):

```
printf ("%d %f %s", i, z, str);
```

In C++ si utilizza la funzione **cout**, che interpreta automaticamente il tipo del dato a partire da quello della variabile da stampare:

```
cout << i << z << str;
```

Lettura di caratteri

La lettura di un carattere singolo, spazi inclusi, si effettua in C con la funzione **getchar()**:

```
char ch;  
ch = getchar();
```

Il corrispondente in C++ è la funzione **cin.get()**:

```
ch = cin.get();
```

Scrittura con formattazione

La stampa sul monitor di un intero *i* in ottale o in esadecimale avviene in C con opportuni caratteri di conversione applicati alla funzione **printf()**:

```
printf ("%o %h", i, i);
```

Fondamenti di Informatica – Input/output standard in C e in C++

Prof. PIER LUCA MONTESSORO - Ing. DAVIDE PIERATTONI
Facoltà di Ingegneria Università degli Studi di Udine

La stessa formattazione si ottiene in C++ con la sequenza:

```
cout << oct << i << hex << i;
```

Scrittura di numeri reali

In C la stampa di un numero reale z può essere effettuata specificando negli argomenti di `printf()` il numero di caratteri totali e il numero di cifre decimali desiderati:

```
printf ("%10.4f", z); /* stampa z su 10 caratteri con 4 cifre decimali */
```

In C++ simili formattazioni, con risultato generalmente diverso, si ottengono ad esempio con:

```
cout << cout.precision(4) << z;
```