



POLITECNICO
DI MILANO

INFORMATICA

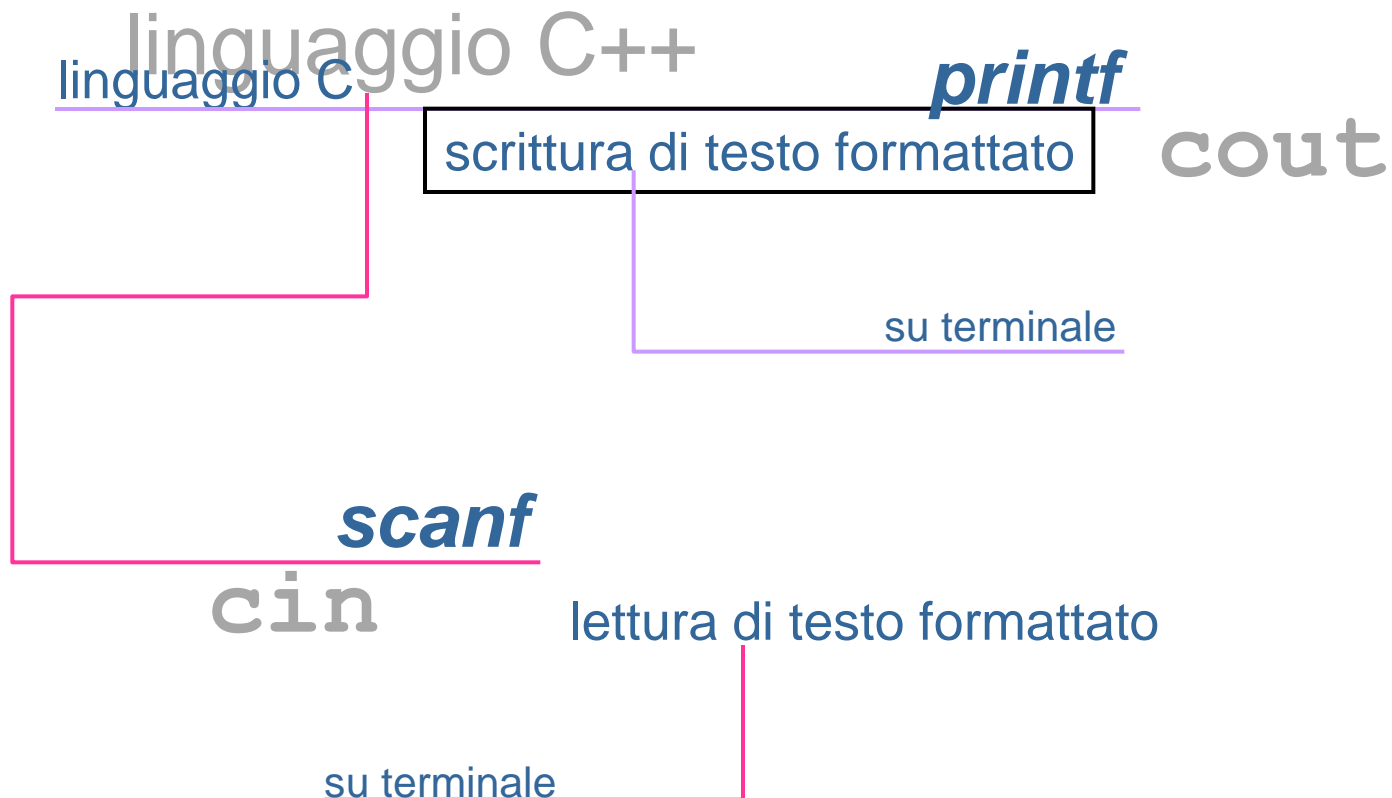
Operazioni di I/O su
terminale nel linguaggio
C

Introduzione

- ✓ Ricordiamo qui alcuni aspetti del linguaggio C, dal quale è stato derivato il linguaggio C++, e che verrà poi in parte utilizzato nei successivi insegnamenti di informatica.
- ✓ Al fine di completare una veloce rassegna sul linguaggio C, dopo aver considerato in dettaglio le caratteristiche del linguaggio C++, è ora necessario considerare i meccanismi standard del linguaggio C per le operazioni di Input/Output.
- ✓ In tale direzione, andremo allora a considerare le funzioni C che svolgono ruoli simili a quelli svolti dalle funzioni cin (scanf/gets), cout (printf/puts) del C++.

I/O su terminale

I/O su terminale



printf

scrive del testo formattato

specifica comandi di controllo

- ritorno a capo
- tabulazione
- nuova linea

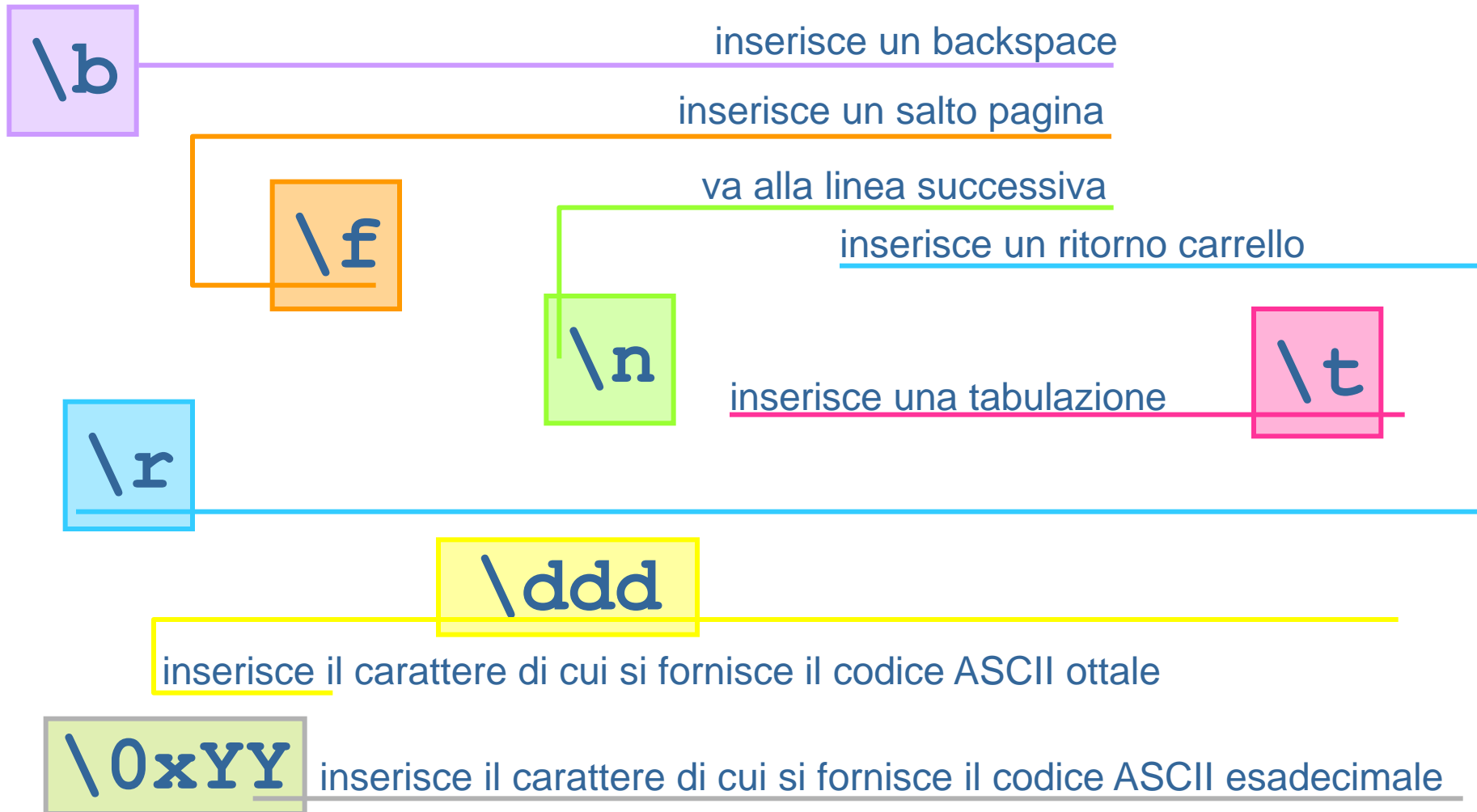
richiede di specificare

formato e variabili

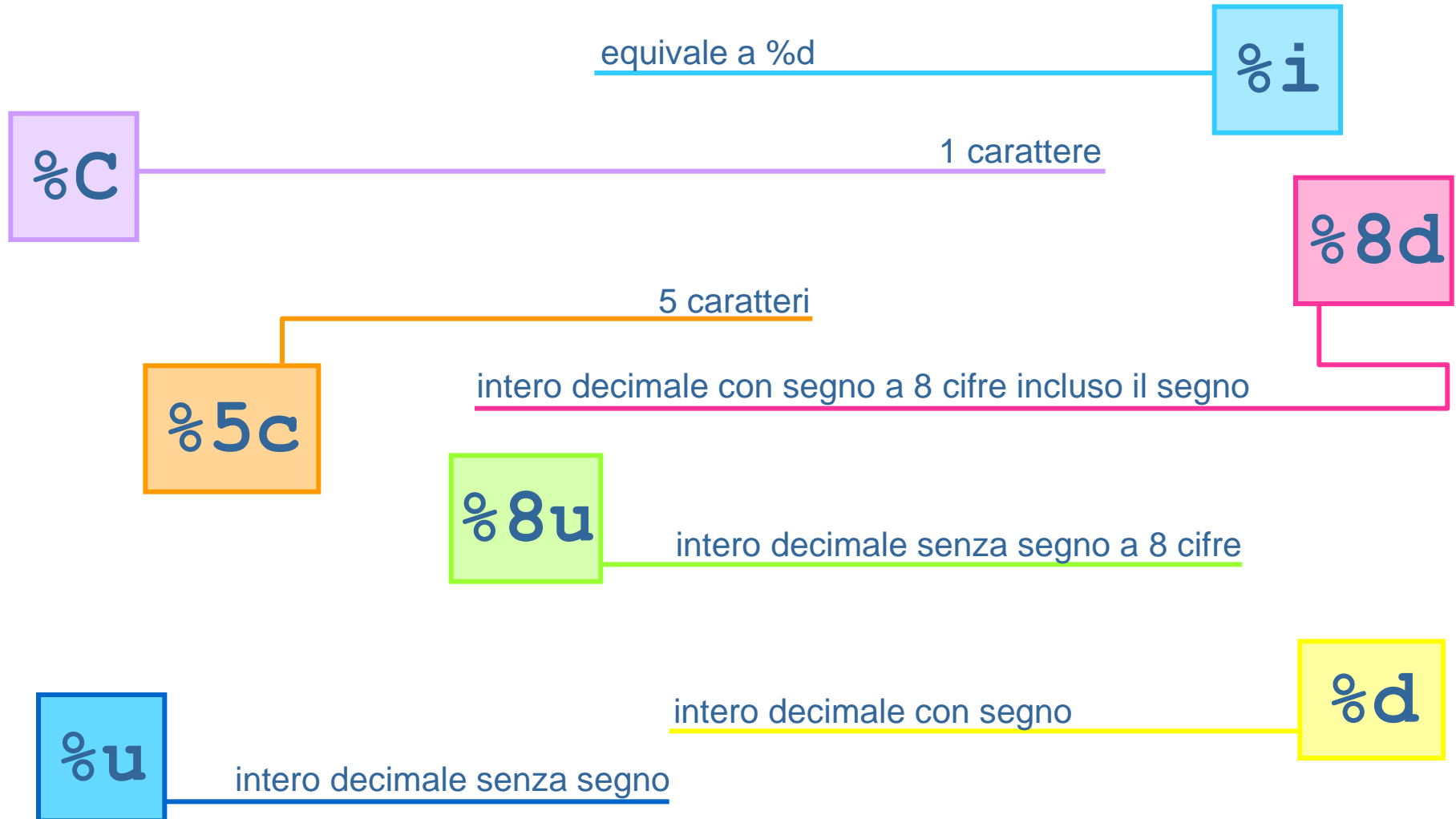
indicare il numero di caratteri
che devono essere utilizzati per la stampa

printf principali comandi di controllo

printf principali comandi di controllo



printf formati per la scrittura di variabili



printf formati per la scrittura di variabili

%s

stringa terminata da '\0'

%40s

i primi 40 caratteri della stringa (o meno se si incontra '\0')

%x

0..9abcdef

intero esadecimale, senza segno

%X

0..9ABCDEF

%o

intero ottale senza segno

printf formati per la scrittura di variabili

%f

un float o un double con segno

%f8.3

un float o un double con segno, su 8 caratteri totali e 3 caratteri decimali

un float o un double con segno in formato esponenziale indicato con **'e'** e con segno

%e

%E

un float o un double con segno in formato esponenziale indicato con **'E'** e con segno

printf allineamento a sinistra

utilizzare il simbolo -

tra il carattere % ed il carattere che **specifica il formato**

%-5d

intero di 5 caratteri allineati a sinistra

scanf

legge del testo formattato

specificare

le variabili che devono essere lette

il formato ed il numero di caratteri che devono essere utilizzati

attraverso il loro puntatore

le variabili lette vengono restituite per indirizzo

`&my_string`

`&my_int`

printf

ipotizziamo di disporre delle seguenti variabili

```
my_string= "la mia stringa";  
my_int = 27;
```

```
printf("una stringa: %s\n ed un intero: %d\n",  
      my_string, my_int);
```

una stringa: la mia stringa
ed un intero: 27

```
printf("un intero: %8d\n ed un reale: %f9.7\n",  
      my_int, atan(1)*4);
```

un intero: 27
ed un reale: 3.1415926

scanf

utilizza i formati della *printf*

equivalente a %d

%c

1 carattere

%i

intero decimale con segno

%d

intero esadecimale

%x

intero ottale

%o

%u

intero decimale senza segno

%e

un float o un double con segno

in formato esponenziale

un float o un double con segno

%f

Estensione di *printf* e *scanf*

scanf

lettura di una stringa

%s

stringa che non include il carattere spazio

i caratteri successivi verranno trascurati

per includere il carattere spazio

gets

Estensione di *printf* e *scanf*

funzioni applicabili senza variazioni anche a

file

```
fprintf(fp, "...", ...) ; /* scrittura su file */
```

```
fscanf(fp, "...", ...) ; /* lettura da file */
```

stringhe

```
sprintf(string_ptr, "...", ...) ;  
/* scrittura su stringa */
```

```
sscanf(string_ptr, "...", ...) ; /* lettura da stringa */
```