

Input e Output

- L'immissione dei dati di un programma e l'uscita dei suoi risultati avvengono attraverso operazioni di **lettura e scrittura**
- **C non ha istruzioni predefinite** per l'input/output
- esiste una **Libreria Standard (stdio)** che mette a disposizione alcune funzioni (dette *funzioni di libreria*) per effettuare l'input e l'output
- Le dichiarazioni delle funzioni messe a disposizione da tale libreria devono essere incluse nel programma:

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

- **#include** è una direttiva per il **preprocessore C**
- nella fase precedente alla compilazione del programma ogni direttiva "#..." viene eseguita, provocando delle modifiche testuali al programma sorgente. Nel caso di **#include <nomefile>** viene sostituita l'istruzione stessa con il contenuto del file specificato
- **Dispositivi standard di input e di output:** generalmente tastiera e video

Input e Output con formato

- Nell'I/O con formato occorre specificare il ***formato*** (il *tipo*) dei dati che si vogliono leggere oppure stampare
- Il ***formato*** stabilisce:
 - **come interpretare** la sequenza dei caratteri immessi dal dispositivo di ingresso (nel caso della lettura)
 - con quale sequenza di caratteri **rappresentare** in uscita i valori da stampare (nel caso di scrittura)

Lettura con formato: scanf

- È una **particolare forma di assegnamento**:
scanf() assegna i valori letti alle variabili specificate come argomenti (nell'ordine di lettura)

```
scanf(<stringa-formato>, <sequenza-variabili>);
```

- Esempio:
int x;
float y;
scanf("%d %f", &x, &y);

scanf()

scanf() legge una serie di valori in base alle specifiche contenute in *<stringa-formato>* e memorizza i valori letti nelle variabili specificate dopo

- restituisce il **numero di valori letti** e memorizzati
- gli **identificatori** delle variabili a cui assegnare i valori sono sempre preceduti dal **simbolo &** (indirizzo della variabile)
- la *<stringa_formato>* può contenere dei caratteri qualsiasi (scartati durante la lettura), che si prevede vengano immessi dall'esterno, insieme ai dati da leggere

```
printf("inserire ore:min:sec\n");
```

```
scanf("%d:%d:%d", &ore, &min, &sec);
```

si richiede che i tre numeri siano immessi separati dal carattere ":"

Scrittura con formato: printf

`printf ()` viene utilizzata per fornire in uscita il valore di una variabile o, più in generale, il risultato di una espressione

- Anche in scrittura è necessario specificare (mediante una *stringa di formato*) il formato dei dati che si vogliono stampare

```
printf (<stringa-formato>, <sequenza-elementi>)
```

- Esempio
- `printf("somma di %d + %f = %f", x, y, x+y);`

printf()

- **printf** scrive una serie di valori in base alle specifiche contenute in *<stringa-formato>*
- I valori visualizzati sono i risultati delle espressioni che compaiono come argomenti
- **printf** restituisce il numero di caratteri scritti
- La stringa di formato della **printf** può contenere sequenze costanti di caratteri da visualizzare

Formati comuni

Formati più comuni

int	%d
float	%f
carattere singolo	%c
stringa di caratteri	%s

Caratteri di controllo

newline	\n
tab	\t
backspace	\b

Per la stampa del carattere ' % ' si usa: %%

Esempio

```
main ()
{
    int k;
    printf("inserire un numero: ");
    scanf("%d", &k); //scansiono un numero e lo memorizzo in k
    printf("Quadrato di %d: %d", k, k*k);
}
```

Output di questo programma:

Inserire un numero: **3**

Quadrato di 3: 9

Esempio

```
scanf("%c%c%c %d %f", &c1,&c2,&c3,&i,&x);  
printf("%c \n %s", c1,"Fine");
```

Se si immettono in input i seguenti dati

ABC 3 7.345

scanf() effettua i seguenti assegnamenti:

'A' → c1

'B' → c2

'C' → c3

3 → i

7.345 → x

L'output della printf è:

A

Fine

Cosa si scrive in output?

```
a=0;
printf("a: %d\n", a);
a=0;
printf("a=1: %d\n", a=1);
a=0;
printf("a==1: %d\n", a==1);
a=0;
printf("a==0: %d\n", a==0);
a=1;
if (a=4)
printf("se a=4, a: %d\n", a);
```

OUTPUT:

a: 0

a=1: 1

a==1: 0

a==0: 1

se a=4, a: 4