

Fondamenti di Informatica e Laboratorio T-AB
Ingegneria dell'Automazione
a.a. 2008/2009

Lab 04

Input/Output

Input e output in C

- Input con formato:

`scanf("stringa formato", lista variabili);`

- Output con formato:

`printf("stringa formato", lista variabili);`

- Tramite la stringa di formato, si specifica "come"
- Tramite i parametri successivi, si specifica "che cosa"

Esempio 1 input output

Esempio 1 – Echo di un numero intero

- Realizzare (cioè scrivere e compilare) un programma che legga da tastiera un numero intero e ne stampi il valore a video (echo)
- pseudo-algoritmo:
 - Leggo da input un numero intero
 - Salvo il numero letto in una variabile apposita
 - Stampo a video il valore della variabile

Esempio 1 - Soluzione

```
#include <stdio.h>
int main()
{
    int value;

    scanf("%d", &value);
    printf("Valore letto:%d\n", value);
    return 0;
}
```

ESEMPIO 2

Scrivere un programma che legge da input un intero e restituisce il quadrato dell'intero

```
#include <stdio.h>
main()
{int k;
scanf ("%d",&k);
printf("Quadrato di %d: %d\n",k,k*k);
}
```

Esempio:

Se in ingresso viene immesso il dato: 3

La `printf` stampa:

```
Quadrato di 3: 9
```

ESEMPIO 3

Rivediamo l'esempio visto inizialmente:

```
/*programma che, letti due numeri a terminale, ne stampa
   la somma*/

#include <stdio.h>

main()
{ int X,Y; /* p. dichiarativa */

  scanf("%d%d",&X,&Y);/*lettura dei due dati*/
  printf("%d",X+Y);/* stampa della loro somma */
}
```

Dati da input i due valori 26 e -32, il programma stampa:

-6

ESEMPIO 4

```
scanf ("%c%c%c%d%f", &c1, &c2, &c3, &i, &x) ;
```

- Se in ingresso vengono dati:

```
ABC 3 7.345
```

- la `scanf` effettua i seguenti assegnamenti:

```
char c1    'A'  
char c2    'B'  
char c3    'C'  
int  i     3  
float x    7.345
```

scanf: STRINGA DI FORMATO

```
#include <stdio.h>
main()
{
    - int intero1, intero2;
    - float reale1;
    - char car1, car2;

    - scanf("%d%d", &intero1, &intero2);
    printf("%d,%d", intero1, intero2);
}
```


`scanf`: **STRINGA DI FORMATO**

```
scanf ("%d%d", &intero1, &intero2);
```

Inserire due interi separati da uno o più spazi:

12 35

12 35

`scanf`: **STRINGA DI FORMATO**

```
scanf ("%d,%d", &intero1, &intero2);
```

Inserire due interi separati da una (e una sola) virgola (eventuali spazi sono scartati):

12,35

12, 35

~~12 35~~

`scanf`: **STRINGA DI FORMATO**

Regole:

la stringa di formato descrive esattamente quello che deve esserci in input

lo spazio bianco viene considerato un separatore e viene scartato

PERO'...

Lo spazio bianco e' a tutti gli effetti un carattere...
quindi nella lettura di caratteri

```
scanf ("%d%c%c", &interol, &car1, &car2);  
printf ("%d,%c,%c", interol, car1, car2);
```

```
12 A B  
12, ,A
```

La scanf ha preso lo spazio come se fosse il
carattere inserito !

UNA SOLUZIONE

Usare un separatore (anche lo spazio stesso)

spazio



```
scanf("%d %c %c",&interol1,&car1,&car2);  
printf("%d,%c,%c",interol1, car1,car2);
```

```
12 A B  
12,A,B
```

Esercizio 2

- Realizzare un programma che legga da input tre caratteri consecutivi e ne stampi il valore a video (echo)

Esercizio 2 - Soluzione

```
#include <stdio.h>
int main()
{
    char c1, c2, c3;

    scanf("%c%c%c", &c1, &c2, &c3);
    printf("Caratteri:%c %c %c\n", c1, c2, c3);
    return 0;
}
```

ESEMPIO 5

```
#include <stdio.h>
main()
{char Nome='A';
char Cognome='C';
printf("%s\n%c. %c. \n%s\n",
      "Programma scritto da:",
      Nome, Cognome, "Fine");
}
```

Stampa:

Programma scritto da:

A. C.

Fine

-

Esempio 6

Esempio:

stampa della codifica (decimale, ottale e esadecimale) di un carattere dato da input.

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

```
main()
```

```
{ char a;
```

```
printf("Inserire un carattere: ");
```

```
scanf("%c",&a);
```

```
printf("\n%c vale %d in decimale, %o in ottale \\  
e %x in hex.\n",a, a, a, a);
```

```
}
```

Effetti dell'esecuzione:

```
Inserire un carattere: A
```

```
A vale 65 in decimale, 101 in ottale e 41 in hex.
```

Esempio 7

- Realizzare un programma che legga da input tre numeri interi e stampi a video la loro somma e la media.

Esempio 7 - Soluzione

```
#include <stdio.h>

int main()
{
    int num1, num2, num3, somma;
    float media;

    scanf("%d%d%d", &num1, &num2, &num3);

    somma = num1 + num2 + num3;
    media = somma / 3.0F;
    printf("Somma:%d\n", somma);
    printf("Media:%f\n", media);
    return 0;
}
```

UN'ALTRA TRAPPOLA

```
printf("Inserire un numero reale: ");  
scanf("%f",&reale1);  
printf("\nInserire un carattere: ");  
scanf("%c",&car1);  
printf("\nLetti: %f,%c",reale1, car1);
```

Questo frammento di programma sembra corretto...

UN'ALTRA TRAPPOLA

...ma il risultato e' questo:

Inserire un numero reale:

12.4

Inserire un carattere:

Letti: 12.400000,

MOTIVO

L' I/O e' bufferizzato: i caratteri letti da tastiera sono memorizzati in un buffer.

In architetture Windows, il tasto di INVIO corrisponde a 2 (DUE!) caratteri (CR LF): il primo è interpretato come separatore, ma il secondo rimane nel buffer ed è preso come carattere inserito dall'utente.

UNA SOLUZIONE

Leggere il carattere "spurio"

```
printf("Inserire un numero reale: ");
scanf("%f",&reale1);
scanf("%*c"); /* letto e buttato via */
printf("\nInserire un carattere: ");
scanf("%c",&car1);
printf("\nLetti: %f,%c",reale1, car1);
```

UNA SECONDA SOLUZIONE

Vuotare il buffer:

```
printf("Inserire un numero reale: ");
scanf("%f",&reale1);
fflush(stdin); /* Si vuota il buffer */
printf("\nInserire un carattere: ");
scanf("%c",&car1);
printf("\nLetti: %f,%c",reale1, car1);
```


PRECISAZIONE

Questo problema si verifica solo con la lettura di caratteri.

Negli altri casi il doppio carattere nel buffer e' considerato come sequenza di separatori e scartato.