Input

- Stringa di formato
  - Spazi e tabulazioni sono ignorati
  - i caratteri normali (non %) devono corrispondere al successivo carattere non bianco in input
  - Le specifiche di conversione riflettono quelle della funzione `printf`
- Conversione per le variabili numeriche
  - gli spazi bianchi precedenti le cifre o il punto decimale sono ignorati
  - le cifre sono accettate fino al carattere di terminazione: `spazio` o `invio`

Input

- Tra % e il carattere di conversione possono essere presenti nell'ordine:

  l'ampiezza massima del campo: un intero decimale

  - se non è presente si assume ampiezza infinita
  - altrimenti, `scanf` legge un numero massimo di caratteri pari all'intero specificato
    - lo spazio bianco non influisce sul conteggio dei caratteri da leggere

  caratteri flag:

  - `l` (elle) tra % e il carattere di conversione indica che il carattere di conversione sarà `f`, `e`, o `E` e l'argomento corrispondente sarà `double` e non `float`
  - `h` tra % e il carattere di conversione indica che il carattere di conversione sarà `d` e l'argomento corrispondente sarà `short int` e non `int`

Memoria di transito della tastiera

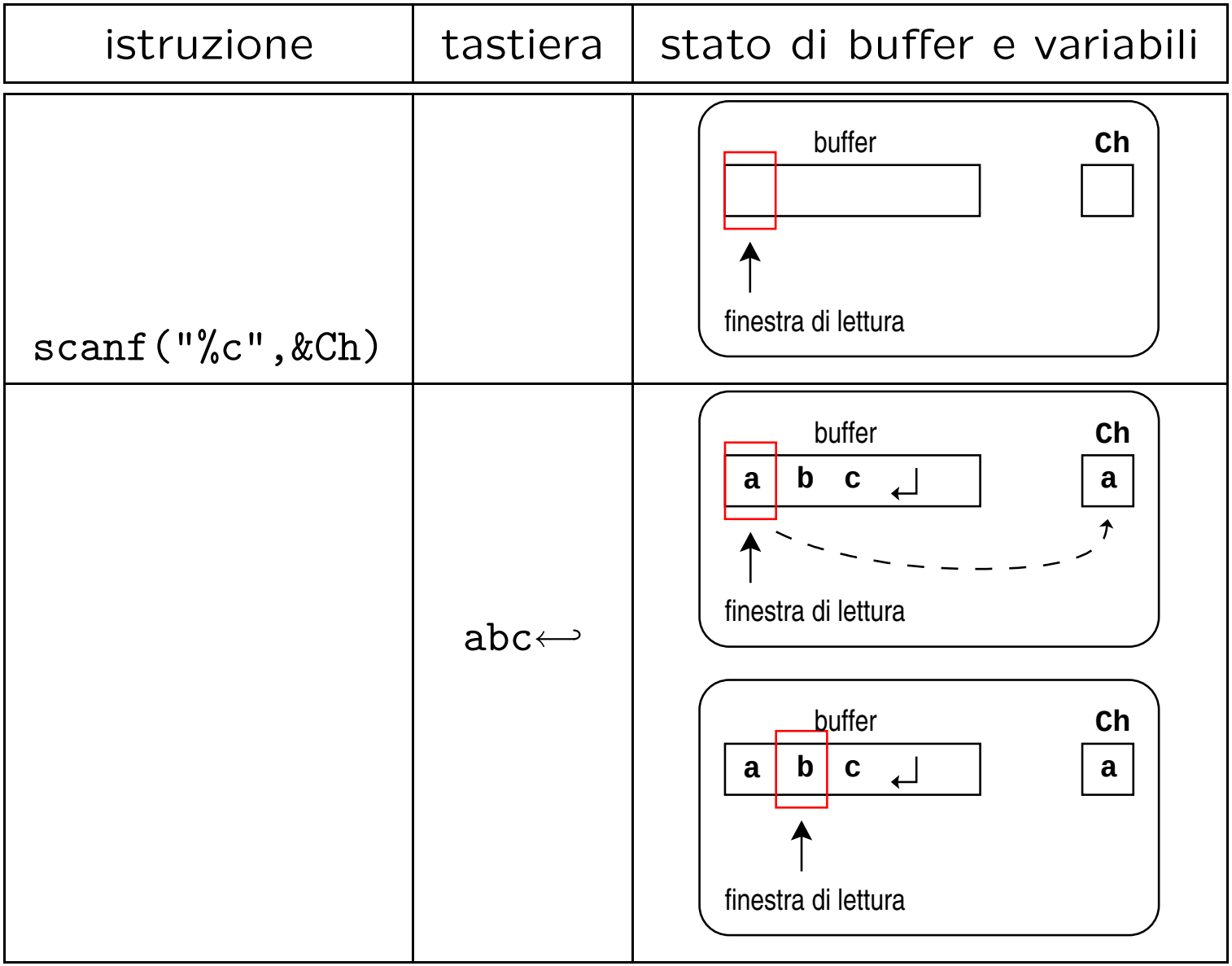
- I caratteri digitati alla tastiera sono memorizzati in un'area di memoria temporanea, la memoria di transito, o buffer, di tastiera
  - Il buffer di tastiera si trova sotto il completo controllo del sistema operativo
- L'esecuzione della funzione `scanf` legge i caratteri da convertire dal buffer di tastiera; se il buffer è vuoto, attende che vengano premuti tasti seguiti da un invio
  - Il numero di caratteri letti in presenza di una data sequenza di caratteri dipende dalla stringa di formato
  - ogni carattere letto è "consumato", nel senso che i caratteri sono letti nella successione in cui si trovano nel buffer (la stessa con cui sono stati digitati)
- Esiste una finestra di lettura che indica la posizione nel buffer del prossimo carattere che viene passato a `scanf`

Memoria di transito della tastiera

Esempio.
```
main() {
    char Ch;
    scanf("%c", &Ch);
}
```

- Inizialmente il buffer è vuoto e `scanf` si arresta per mancanza di caratteri da convertire
- L'utente digita `abc` e i caratteri sono memorizzati nel buffer nell'ordine in cui sono digitati
- Ora la finestra è sul primo carattere `a`
- L'utente digita $\hookleftarrow$ e il primo carattere `a` è consumato da `scanf`; il carattere `'a'` è memorizzato nella variabile `Ch`
- Per effetto della lettura, la finestra avanza al carattere successivo
- Ogni successiva lettura riceverà `b` come primo carattere letto

Memoria di transito della tastiera



Memoria di transito della tastiera

Osservazione (Lettura di un singolo carattere).

```
#include <stdio.h>
main () {
  int I;
  char C;
  printf("Inserire intero e premere enter ->");
  scanf("%d",&I);
  printf("Inserire carattere e premere enter ->");
  scanf("%c" ,&C);
  printf("\nIntero %5d Carattere %c (ASCII %d)",I,C,C);
}
```

- Input: 123↩
- Output:

```
Intero   123 Carattere
 (ASCII 10)
```

- ?