

Problema: buffer di tastiera

Quando il tuo programma esegue un'istruzione di lettura da tastiera, i dati letti vengono depositati nel buffer di tastiera (area di memoria associata alla tastiera). La funzione utilizzata per la lettura (ad es. `scanf()`) legge i dati dal buffer, li inserisce nella variabile che le hai passato come parametro e cancella i dati letti dal buffer stesso.

In generale al termine dell'esecuzione della funzione di libreria che legge i dati, il buffer è vuoto.

In alcuni casi può capitare che il buffer non venga svuotato. Questo crea problemi quando verranno eseguite successive letture da tastiera.

Facciamo un esempio:

Siano `car1` e `car2` due variabili di tipo `char`.

```
printf("Digita un carattere")
scanf("%c", &car1);
```

...

```
printf("Digita un carattere")
scanf("%c", &car2);
```

Che cosa succede esecuzione?

L'esecutore legge dal buffer il primo dato (carattere) e lo mette nella variabile `car1`, poi elimina dal buffer il dato letto.

Quando tu leggi un carattere nel buffer non c'è solo il carattere letto, ma il carattere letto + il carattere di invio (`\n`). Quindi una volta che viene eliminato dal buffer il carattere letto, nel buffer rimane il carattere di invio che ha codice ASCII 10. Quando l'esecutore eseguirà la lettura successiva del `car2` non ti farà inerire nulla, è come se l'istruzione `scanf("%c", &car2);` non venisse eseguita. In realtà, siccome il buffer non è vuoto, l'esecutore legge il dato contenuto nel buffer (carattere invio, codice 10) e lo assegna alla variabile `car2`.

Se successivamente stampi il due caratteri con codice **%d** ti accorgi di quanto è stato fatto.

Esempio:

```
int main(){
char car1, car2 ;

    cout<<"Digita un carattere "<<endl;
    scanf("%c",&car1);

    cout<<"Digita un carattere "<<endl;
    scanf("%c",&car2);

    cout<<"Output"<<endl;
    printf("%d %d",car1,car2);           //%c se vuoi il carattere

    return 0;
}
```

Digita un carattere

A

Digita un carattere

Output

65 10

Process exited after 3.676 seconds with return value 0

Premere un tasto per continuare . . .

Soluzione:

```
int main(){
char car1, car2 ;

    cout<<"Digita un carattere "<<endl;
    scanf("%c",&car1);
    while(getchar() !='\n') ;           //per la pulizia del buffer

    cout<<"Digita un carattere "<<endl;
    scanf("%c",&car2);
    while(getchar() !='\n') ;           //per la pulizia del buffer

    cout<<"Output"<<endl;
    printf("%d %d",car1,car2);          //%c se vuoi il carattere

    return 0;
}
```

Digita un carattere

A

Digita un carattere

B

Output

65 66

Process exited after 4.258 seconds with return value 0

Premere un tasto per continuare . . .

Conclusioni

Quando, mandando in esecuzione un programma, ti accordi che viene “saltata” una istruzione di lettura (in realtà essa viene eseguita!) devi inserire dopo ogni istruzione di lettura un’istruzione che pulisce il buffer.

In C:

```
while(getchar() != '\n');
```

oppure

```
fflush(stdin);
```

In C++:

```
while(cin.get() != '\n');
```