**8° lezione prog1**

**Famiglia di funzioni scanf()**

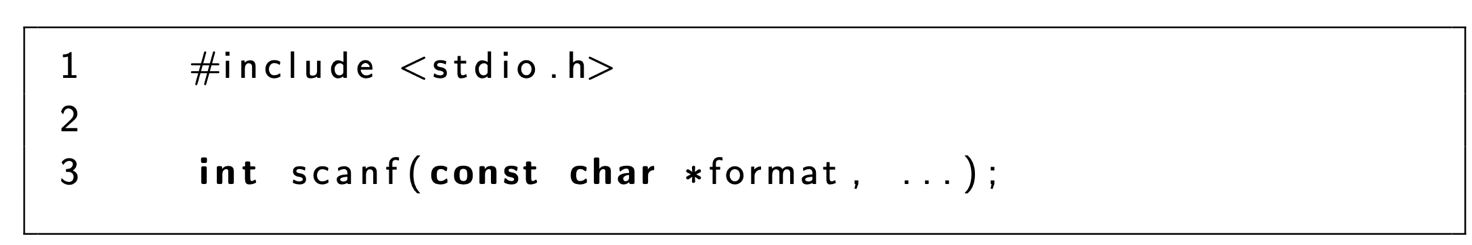
Le funzioni della famiglia scanf() permettono di eseguire l’input.

* **La funzione scanf()** permette di leggere l’input formattato dallo standard input.
* **La funzione fscanf()** permette di leggere byte a partire dallo da uno stream che viene specificato all’intero della funzione stessa.

Per invocare leggere la documentazione dal terminale (su piattaforma UNIX LIKE) bisogna scrivere il seguente comando: **man scanf.**

Per l’utilizzo di queste famiglie di funzioni bisogna includere la libreria standard I/O (stdio.h)

**La funzione scanf()**



Questa funzione legge dallo standard input, bisogna specificare un numero variabile di parametri(ellipsi numero variabili di parametri) e una stringa di formato (specificatori di conversione), per indicare al compitore quale tipo di dato in input si deve aspettare.

**Specifica di conversione**

* Iniziano con %
* Modificatore di ampiezza cioè max caratteri da leggere
* Modificatore di lunghezza (l per i long)
* Specificatole di conversione (%d,%f,%c)

**Ha la seguente sintassi: scanf(“%d”,&n);**

**Il simbolo & (undrsend) indica l’indirizzo in memoria della variabile da prendere in input. Nello scanf() per le variabili si utilizza per array no in quanto l’identificatore dell’array è un puntatore che rappresenta l’indirizzo (1° elemento dell’array) solo nelle stringhe.** Perché nel caso delle stringhe sto acquisendo tutta la stringa e non un singolo numero**.**

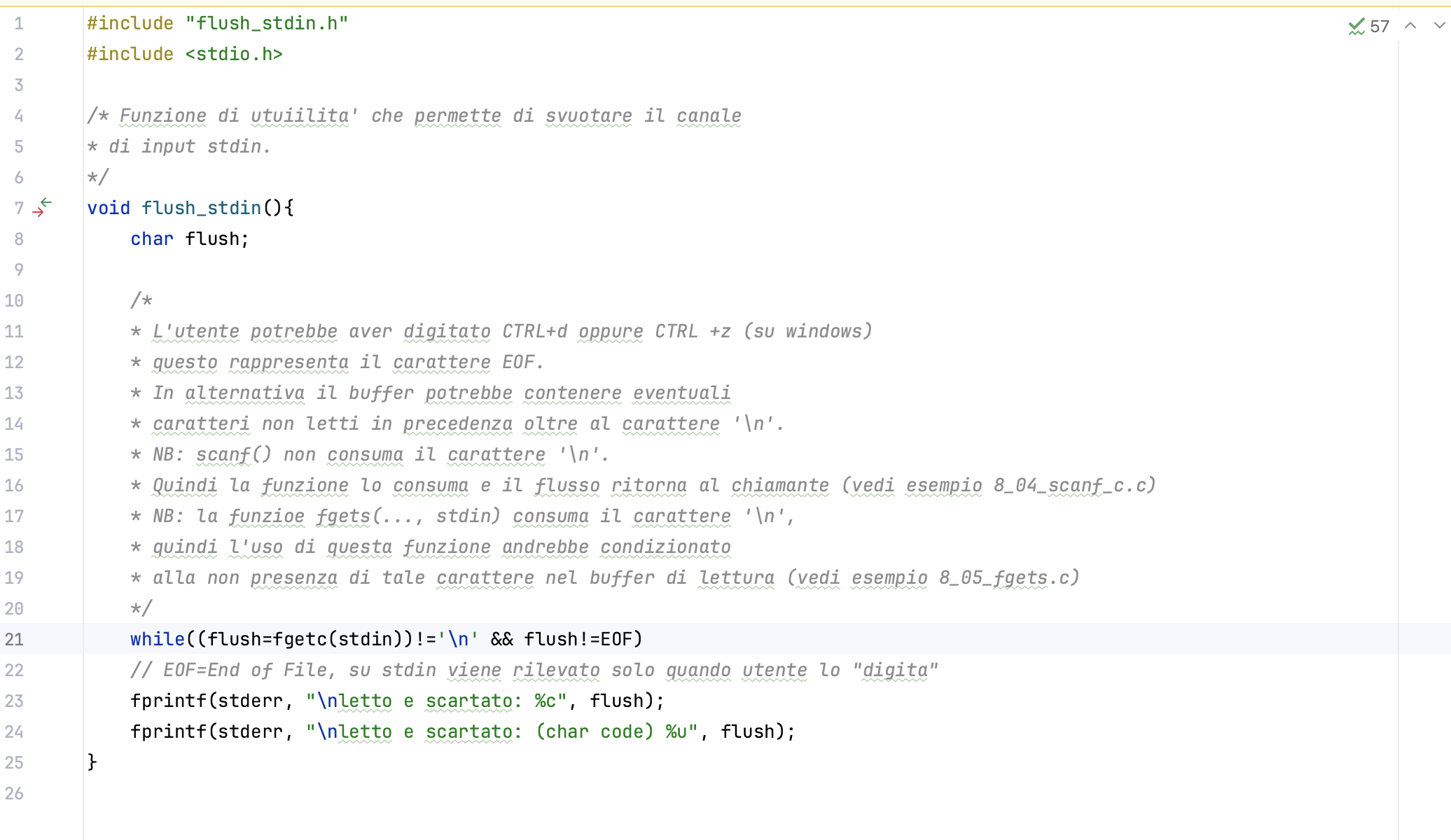
Questa funzione restituisce il numero di tocken(valori) presi in input.

L’ EOF(End Of File) per lo standard input si ottiene digitando ctrl-d o ctrl-z su Windows. Dopo aver digitato questa combinazione di tasti lo stream non è più utilizzabile pertanto tutti gli altri tentavi di lettura mediante scanf() restituiranno EOF. Viene realizzato in questo modo perché se l’utente forza l’EOF significa che non vuole più inserire dati da tastiera e quindi il processo di acquisizione viene interrotto.

Le stringhe in C sono array (vettori) di caratteri. Char nome[20]; questo array potrà contenere 19 caratteri in quanto il ventesimo è occupato da ‘/0’ (carattere di fine stringa o anche detto “tappo”) in codice ASCII 0. La funzione scanf() quando legge una stringa lo copia in un array di caratteri ed alla fine inserisce il carattere di fine stringa(che non viene inserito dall’ utente ma dalla funzione stessa).

Se dopo una qualunque acquisizione si chiede un carattere in input si va a prendere lo \n o anche altri caratteri(perché se ad esempio si crea una stringa di 10 caratteri e ne inseriamo 20 i rimanenti 10 che non sono stanti inseriti rimano nel buffer) che indica la fine dell’acquisizione del dato preso in precedenza. Per tale motivo si ha la possibilità di inserire il nuovo carattere (il buffer è ha un carattere che non è stato utilizzato e quindi lo inserisce). Per evitare ciò bisogna svotare il buffer. Per non superare la dimensione dell’array di caratteri nello scanf() possiamo inserire un modificatore di lunghezza. (NB vedere esempio proposto dal prof)

**Attenzione per acquisire un doble lo specificatore di conversione nello scanf()è %lf.**



Nell’immagine soprastante si ha l’implementazione di una funzione per lo svotamento del buffer.

Nel file header(.h) sono presenti delle direttive al pre-processore ed un prototipo della procedura(stampino della funzione dove si indicano i parametri in ingresso e quello di ritorno/ uscita (void) parametri formali non cioè che non hanno valore)

* fgetc() è una funzione che legge un carattere dallo stream passato tra parentesi tonde come parametro attuale e lo restituisce al chiamante, pertanto il valore restituito dalla funzione deve essere immagazzinato in una variabile..
* #ifndef FLUSH: questa direttiva chiede al micro-processore se non è stata definita una direttiva di nome FLUSH. Se non è stata definita lo si fa con la seguente istruzione: #define FLUSH. Inoltre si va ad definire il prototipo della procedura, dopo si indica il marcatore di fine if scrivendo: #endif. Per include questo file header/libreria in un file si utilizza la seguente istruzione: #include “flush\_stdin.h”. In questo modo si va a fornire al compilatore la funzione.

Con la direttiva condizionale la prima traslation unit che viene processata dal microprocessore includerà tutte le varie definizioni e verrà anche definito FLUSH e quando il microprocessore analizzerà le altre traslation unit le direttive e la procedure dentro #ifndef non saranno più incluse. Se non si utilizza questa procedura si avrebbe un errore in quanto più traslation unit includeranno le direttive e la procedura (saranno definiti più volte). Se si vuole utilizzare la libreria appena creato si deve utilizazre #include “flush\_stdin.h” in quanto le “” indicano al compilatore che il file header non son si trova all’interno di locazioni standard ma si trova all’interno della stessa directory(cartella) in cui si trova il file dove si sta includendo questo file.

Per generare un file eseguibile con dui file si utilizza il seguente comando: gcc file1.c file2.c.

**Le funzione fscanf() sscanf() e fgets()**

Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente

* **La funzione fscanf()** è come la scanf() ma legge dalllo stream specificato come primo parametro (restituisce il numero di token)
* **La funzione sscanf()** è come la funzione scanf() con la differenza che permette di prelevare i dati dal buffer.
* **La funzione fgetc()** permette di leggere un certo numero di caratteri dallo stream e di memorizzali in byte e ritorna il numero di caratteri letti. È come lo scanf() con la differenza che permette di scartare ulteriori caratteri rimasti nel buffer. (NB vedere esempio proposto dal prof)

**Costrutti condizionali del C per il controllo del flusso**

**La selezione semplice**

La selezione semplice permette di far eseguire alcune istruzioni solamente quando il valore di un’istruzione di test (o condizione logica) risulta vera nel caso falso non viene eseguita nessuna istruzione.

Una condizione è un’espressione del linguaggio rappresentata da due elementi messi a confronto da un operatore relazionale

Gli operatori relazionali

|  |  |  |
| --- | --- | --- |
| Operatore | Significato | Esempio |
| > | Maggiore | a>b |
| < | Minore | a<b |
| >= | Maggiore o uguale | a>=b |
| <= | Minore o uguale | a<=b |
| == | Uguale | a==b |
| != | Diverso | a!=b |

Attenzione a non confondere == con = in quando il prima si utilizza in una condizione il secondo si utilizza per assegnare un valore ad una variabile(int a=10;). Nel caso in cui nell’ if si commettesse questo errore si avrebbe come risultato true tranne se il nuomero è 0

Sintassi  if struttura semplice

If (espessione o predicato)

{

Istruzioni;

}

La selezione doppia

L’istruzione di sezione di selezione doppia permette di eseguire alternativamente alcune istruzioni, infatti, se la condizione è vera vengono eseguite le istruzioni nel ramo di sinistra invece se è falsa vengono eseguite le istruzioni di destra.

Immagine che contiene diagramma, testo, linea, simbolo

Descrizione generata automaticamente

La sintassi della selezione doppia è la seguente:

if(condizione o predicato)

{

Istruzioni;

}

}

else

{

Istruzioni;

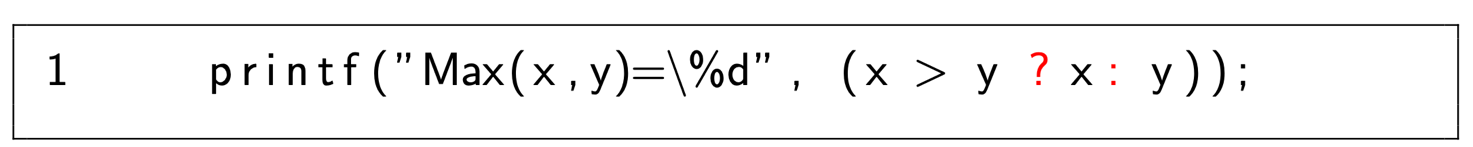
}

Se si ha solo una sola istruzione si può evitare di inserire le { }. Se si commette questo errore verrà eseguita solo la prima istruzione e le altre verranno eseguite sempre indipendentemente dalla condizione. Se si dimenticano le partesi {} nella selezione doppia si ha un errore in quando l’else solo non può stare, deve essere accoppiato all’ if.

**Operatore condizionale**

Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente



L’istruzione switch-case

Serve a codificare gli algoritmi in cui ci sono piu controlli sulla stessa variabile. Sono controlli di uguaglianza e non di maggioranza o minoranza.

Si sostituisce cosi la serie di if nidificati.

All’interno dello switch si fanno tutti i casi il break serve a stoppare lo switch cioè ad uscire fuori dallo switch una volta trovato il simbolo scelto dall’utente

La struttura è la seguente

switch(espressione)

{

case valore1:{

istruzioni;

break;

}

case valore2:{

istruzioni;

break;

}

default

{

istruzioni;

break;

}

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Situazione particolare i cui i break; non si inseriscono. In questo caso dal case1 fino al 3 non si fa niente, invece ,al case 4 si esegue il printf().