

## LA LIBRERIA STANDARD DEL C

---

- La *libreria standard* del C è in realtà *un insieme di librerie*
- Per usare una libreria, *non occorre inserirla esplicitamente nel progetto*: ogni ambiente di sviluppo *sa già* dove cercarle
  - Rhide: C:\DJGPP\LIB
  - Turbo C: vedi *Options/Project/Directories*
- Ogni file sorgente che ne faccia uso deve però *includere lo header opportuno*, che contiene le dichiarazioni necessarie.

## LA LIBRERIA STANDARD DEL C

---

### Le librerie standard

- |                                   |                       |
|-----------------------------------|-----------------------|
| • <b>input/output</b>             | <b><i>stdio.h</i></b> |
| • funzioni matematiche            | <i>math.h</i>         |
| • gestione di stringhe            | <i>string.h</i>       |
| • operazioni su caratteri         | <i>ctype.h</i>        |
| • gestione dinamica della memoria | <i>stdlib.h</i>       |
| • ricerca ed ordinamento          | <i>stdlib.h</i>       |
| • ...                             |                       |
| • ... e molte altre.              |                       |

## LA LIBRERIA `stdio`

---

- L'input avviene di norma dal **canale standard di input (*stdin*)**
- L'output avviene di norma sul **canale standard di output (*stdout*)**
- input e output avvengono sotto forma di una *sequenza di caratteri*
- tale sequenza è terminata dallo speciale carattere EOF (*End-Of-File*), che varia da una piattaforma all'altra.

## I CANALI STANDARD

---

Di norma:

- il canale standard di input, *stdin*, coincide con la tastiera
- il canale standard di output, *stdout*, coincide con il video

Esiste inoltre un altro canale di output, riservato ai messaggi di errore: *stderr*.

- anch'esso di norma coincide con il video

*Avere un canale separato per i messaggi di errore è utile se *stdout* viene ridiretto su un altro dispositivo fisico (stampante, modem, etc): così i messaggi d'errore rimangono sul video.*

## INPUT OUTPUT A CARATTERI

---

- Poiché sui canali di I/O fluiscono *sequenze di caratteri*, esistono *due operazioni base*:
- *scrivere un carattere sul canale di output*  
`putchar(ch);`
- *leggere un carattere dal canale di input*  
`ch = getchar();`
- Ogni altro tipo di I/O può essere costruito a partire da queste *operazioni primitive*.

## INPUT OUTPUT A CARATTERI

---

- `int putchar(int ch);`
  - scrive un carattere sul canale di output
  - restituisce il carattere scritto, o EOF in caso di errore.
- `int getchar(void);`
  - legge un carattere dal canale di input
  - restituisce in carattere letto, o EOF in caso la sequenza di input sia finita, o in caso di errore.

*Entrambe le funzioni leggono/scrivono un carattere convertito in int (senza estensione di segno): ciò assicura che il valore numerico del carattere non sia mai negativo (escluso, eventualmente, il solo carattere speciale EOF).*

## ESEMPIO

---

Ricopiare l'input standard sull'output standard, carattere per carattere.

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

```
main(){
```

```
    int c;
```

```
    while(( c=getchar() ) != EOF)
```

```
        putchar(c);
```

```
}
```

**Attenzione:** `getchar()` inizia a produrre caratteri solo *dopo* aver premuto INVIO.

*Per chiudere l'input producendo un EOF da tastiera, CTRL+Z in sistemi Windows, CTRL+D in Unix.*