

Stringhe

# Le stringhe

- Le stringhe sono sequenze di caratteri,
  - in C le stringhe costanti vengono denotate da una successione di caratteri racchiusa fra apici

Es:

`"ciccio"`

`"n = %d"`

`"Ciao Mondo"`

- La rappresentazione interna è come un array di caratteri non modificabile terminato dal carattere `'\0'`

c	i	c	c	i	o	\0
---	---	---	---	---	---	----

# Le stringhe

c	i	c	c	i	o	\0
---	---	---	---	---	---	----

- Quindi una stringa occupa un array con un carattere in più riservato al carattere terminatore
- Le stringhe variabili sono rappresentate come array di caratteri:  
**char parola[M], frase[N];**
- Per quanto detto prima bisogna sempre ricordarsi di allocare un carattere in più dei caratteri contenuti nelle stringhe che intendiamo scriverci dentro

# Le stringhe

Per leggere & stampare una stringa si utilizzano `printf()` e `scanf()` con il modificatore `%s`

- As esempio:

```
int main (void) {  
    char parola[M];  
    printf("Inserisci una stringa:\n");  
    scanf("%s",parola);  
    printf("Ho letto: %s\n",parola);  
    return 0;  
}
```

# Le stringhe

```
$ ./esegui
```

```
Inserisci una stringa:
```

Se digito `ciccio` e `newline` ↓

```
$ ./esegui
```

```
Inserisci una stringa: ciccio
```

```
Ho letto ciccio
```

```
$
```

# Le stringhe

Per stampare una stringa si utilizzano `printf()` e `scanf()` con il modificatore `%s`

- As esempio:

```
int main (void) {  
    char parola[M];  
    printf("Inserisci una stringa:\n");  
    scanf("%s", parola);  
    printf("Ho letto: %s\n", parola);  
    return 0;  
}
```

Non dobbiamo mettere `&`  
perchè è un array  
(vedi più avanti passaggio  
di array a funzioni)

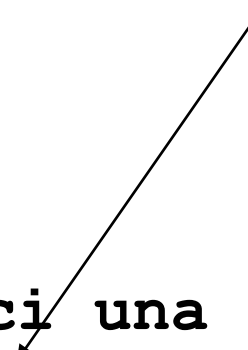
# Le stringhe

Per stampare una stringa si utilizzano `printf()` e `scanf()` con il modificatore `%s`

- As esempio:

```
int main (void) {  
    char parola[M];  
    printf("Inserisci una stringa:\n");  
    scanf("%s", parola);  
    printf("Ho letto: %s\n", parola);  
    return 0;  
}
```

`scanf()` legge tutta la stringa fino al primo spazio `\t` o `\n` e la scrive in parola anche se ha letto più di `M-1` char



# Le stringhe

- La libreria **string.h** contiene un insieme di funzioni predefinite per lavorare con le stringhe, ad esempio
  - **strcpy, strncpy()** per copiare una stringa in un'altra
  - **strcmp(), strncmp()** per confrontarle lessicograficamente
  - **strcat, strncat()** per concatenarle
  - **strlen()** per avere la lunghezza (senza il terminatore)
  - E molto altro

Si tratta di funzioni utili ma che possono avere dei **malfunzionamenti anche gravi se non usate bene** descriveremo in dettaglio caratteristiche e problemi più avanti, per il momento prima di usarle leggete attentamente il manuale (Sezione 3)