



# Programmazione I

## Il Linguaggio C

### Le stringhe

Daniel Riccio

Università di Napoli, Federico II

27 ottobre 2021



# Sommario



- Argomenti
  - Funzioni sulle stringhe
  - Vettori di stringhe
  - Esercizi

# Funzioni di libreria

Quando possibile, utilizzare sempre le funzioni di libreria

Sono più veloci

Sono maggiormente collaudate

In ogni caso, ricordare che è sempre possibile effettuare le operazioni direttamente:

Sui caratteri, ricorrendo alla codifica ASCII

Sulle stringhe, ricorrendo alla rappresentazione come vettori di caratteri

# Funzioni sui caratteri



Definite in <ctype.h>

Analizzano un singolo carattere,  
identificandone la tipologia

Lettera

Maiuscola

Minuscola

Cifra

Punteggiatura

**isalpha**

**isupper**

**islower**

**isdigit**

**isalnum**

**isxdigit**

**ispunct**

**isgraph**

**isprint**

**isspace**

**iscntrl**

# Funzioni isalpha/isdigit



Nome funzione	<b>isalpha</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è una lettera maiuscola o minuscola (A...Z, a...z), "falso" altrimenti
Esempio	<pre>if(isalpha(ch)) { ... }</pre>

Nome funzione	<b>isdigit</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è una cifra numerica (0...9), "falso" altrimenti
Esempio	<pre>if(isdigit(ch)) { ... }</pre>

# Funzioni isupper/islower



Nome funzione	<b>isupper</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è una lettera maiuscola (A...Z), "falso" altrimenti
Esempio	<pre>if(isupper(ch)) { ... }</pre>

Nome funzione	<b>islower</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è una lettera minuscola (a...z), "falso" altrimenti
Esempio	<pre>if(islower(ch)) { ... }</pre>

# Funzioni isalnum/isdigit



Nome funzione	<b>isalnum</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è una lettera oppure una cifra numerica, "falso" altrimenti. Equivalente a <code>isalpha(ch)    isdigit(ch)</code>
Esempio	<code>if(isalnum(ch))</code> <code>{ ... }</code>

17

Nome funzione	<b>isxdigit</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è una cifra numerica oppure una lettera valida in base 16 (a...f, A...F), "falso" altrimenti.
Esempio	<code>if(isxdigit(ch))</code> <code>{ ... }</code>

# Funzioni ispunct/isgraph



Nome funzione	<b>ispunct</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è un simbolo di punteggiatura (!"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~), "falso" altrimenti.
Esempio	<pre>if(ispunct(ch)) { ... }</pre>

Nome funzione	<b>isgraph</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è un qualsiasi simbolo visibile (lettera, cifra, punteggiatura), "falso" altrimenti.
Esempio	<pre>if(isgraph(ch)) { ... }</pre>



# Funzioni isprint/isspace



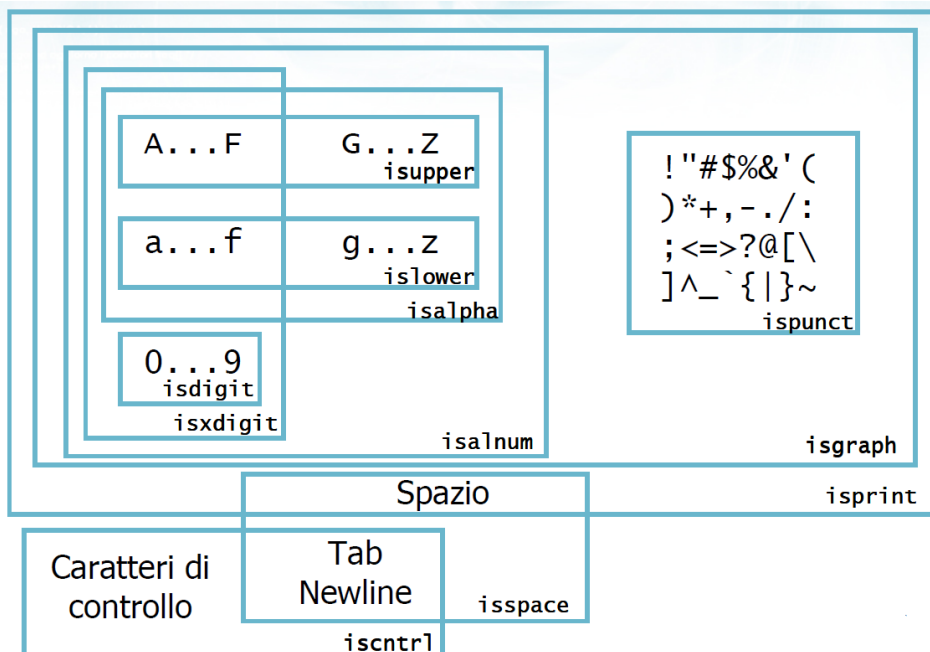
Nome funzione	<b>isprint</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è un qualsiasi simbolo visibile oppure lo spazio, "falso" altrimenti.
Esempio	<pre>if(isprint(ch)) { ... }</pre>

Nome funzione	<b>isspace</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se il carattere ch è invisibile (spazio, tab, a capo), "falso" altrimenti.
Esempio	<pre>if(isspace(ch)) { ... }</pre>

# Funzione iscntrl



Nome funzione	<b>iscntrl</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	Vero/falso
Descrizione	Restituisce "vero" se ch è un carattere di controllo (ASCII 0...31, 127), "falso" altrimenti.
Esempio	<pre>if(iscntrl(ch)) { ... }</pre>



# Funzioni toupper/tolower



Nome funzione	<b>toupper</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	<b>char</b> : carattere maiuscolo
Descrizione	Se ch è una lettera minuscola, <i>restituisce</i> l'equivalente carattere maiuscolo, se no <i>restituisce</i> ch stesso
Esempio	<pre>for(i=0; s[i]!=0; i++)     s[i] = toupper(s[i]) ;</pre>

Nome funzione	<b>tolower</b>
Libreria	<code>#include &lt;ctype.h&gt;</code>
Parametri in ingresso	ch : carattere
Valore restituito	<b>char</b> : carattere maiuscolo
Descrizione	Se ch è una lettera minuscola, <i>restituisce</i> l'equivalente carattere maiuscolo, se no <i>restituisce</i> ch stesso
Esempio	<pre>for(i=0; s[i]!=0; i++)     s[i] = tolower(s[i]) ;</pre>

# Funzioni strcpy/strncpy



Nome funzione	<b>strcpy</b>
Libreria	<code>#include &lt;string.h&gt;</code>
Parametri in ingresso	<code>dst : stringa</code> <code>src : stringa</code>
Valore restituito	nessuno utile
Descrizione	Copia il contenuto della stringa <code>src</code> all'interno della stringa <code>dst</code> (che deve avere lunghezza sufficiente).
Esempio	<code>strcpy(s1, s2) ;</code> <code>strcpy(s, "") ;</code> <code>strcpy(s1, "ciao") ;</code>

Nome funzione	<b>strncpy</b>
Libreria	<code>#include &lt;string.h&gt;</code>
Parametri in ingresso	<code>dst : stringa</code> <code>src : stringa</code> <code>n : numero max caratteri</code>
Valore restituito	nessuno utile
Descrizione	Copia il contenuto della stringa <code>src</code> (massimo <code>n</code> caratteri) all'interno della stringa <code>dst</code> .
Esempio	<code>strncpy(s1, s2, 20) ;</code> <code>strncpy(s1, s2, MAX) ;</code>



# Funzioni strcat/strncat

Nome funzione	<b>strcat</b>
Libreria	<code>#include &lt;string.h&gt;</code>
Parametri in ingresso	<code>dst : stringa</code> <code>src : stringa</code>
Valore restituito	nessuno utile
Descrizione	Accoda il contenuto della stringa <code>src</code> alla fine della stringa <code>dst</code> (che deve avere lunghezza sufficiente).
Esempio	<code>strcat(s1, s2) ;</code> <code>strcat(s1, " ") ;</code>

Nome funzione	<b>strncat</b>
Libreria	<code>#include &lt;string.h&gt;</code>
Parametri in ingresso	<code>dst : stringa</code> <code>src : stringa</code> <code>n : numero max caratteri</code>
Valore restituito	nessuno utile
Descrizione	Accoda il contenuto della stringa <code>src</code> (massimo <code>n</code> caratteri) alla fine della stringa <code>dst</code> .
Esempio	<code>strncat(s1, s2) ;</code>

34

# Funzioni strcmp/strncmp



Nome funzione	<b>strcmp</b>
Libreria	<code>#include &lt;string.h&gt;</code>
Parametri in ingresso	s1 : stringa s2 : stringa
Valore restituito	<b>int</b> : risultato confronto
Descrizione	Risultato <0 se s1 precede s2 Risultato ==0 se s1 è uguale a s2 Risultato >0 se s1 segue s2
Esempio	<pre>if(strcmp(s, r)==0) {...} while(strcmp(r,"fine")!=0) {...}</pre>

37

Nome funzione	<b>strncmp</b>
Libreria	<code>#include &lt;string.h&gt;</code>
Parametri in ingresso	s1 : stringa s2 : stringa n : numero max caratteri
Valore restituito	<b>int</b> : risultato confronto
Descrizione	Simile a strcmp, ma confronta solo i primi n caratteri, ignorando i successivi.
Esempio	<pre>if(strncmp(r, "buon", 4)==0) (buongiorno, buonasera, buonanotte)</pre>



# Funzioni atoi/atof



Nome funzione	<b>atoi</b>
Libreria	<code>#include &lt;stdlib.h&gt;</code>
Parametri in ingresso	<code>s</code> : stringa
Valore restituito	<b>int</b> : valore estratto
Descrizione	Analizza la stringa <code>s</code> ed estrae il valore intero in essa contenuto (a partire dai primi caratteri).
Esempio	<code>n = atoi(s) ;</code> <code>n = atoi("232abc") ;</code>

Nome funzione	<b>atof</b>
Libreria	<code>#include &lt;stdlib.h&gt;</code>
Parametri in ingresso	<code>s</code> : stringa
Valore restituito	<b>double/float</b> : valore estratto
Descrizione	Analizza la stringa <code>s</code> ed estrae il valore reale in essa contenuto (a partire dai primi caratteri).
Esempio	<code>x = atof(s) ;</code> <code>x = atof("2.32abc") ;</code>

# Esercizio – Parola palindroma

Sia data una parola inserita da tastiera.

Si consideri che la parola può contenere sia caratteri maiuscoli che caratteri minuscoli, e complessivamente al massimo **30** caratteri

Il programma deve svolgere le seguenti operazioni:

- Visualizzare la parola inserita

- Verificare se la parola è palindroma

o	t	t	o
---	---	---	---

m	a	d	a	m
---	---	---	---	---



# Esercizio – Parola palindroma



Acquisisci parola

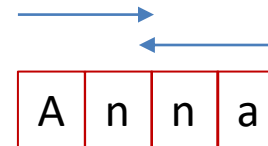
```
#define MAX 30  
char parola[MAX+1];  
printf("Inserisci parola: ");  
scanf("%s", parola);
```

Stampa parola

```
printf("Parola inserita: %s\n", parola);
```

Verifica se è palindroma

```
int i, j, palin, lun;  
i = 0;  
j = strlen(parola) - 1;  
palin = 1;  
while( i < j && palin == 1 ){  
    if(parola[i] != parola[j])  
        palin = 0;  
    i++; j--;  
}
```



È corretto?



# Esercizio – Parola palindroma



Acquisisci parola

```
#define MAX 30
char parola[MAX+1];
printf("Inserisci parola: ");
scanf("%s", parola);
```

Stampa parola

```
printf("Parola inserita: %s\n", parola);
```

Converti in minuscolo

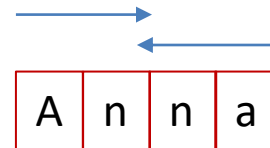
```
char minusc[MAX+1];
int i;
strcpy(minusc, parola);
for(i=0; minusc[i]!=0; i++){
    minusc[i] = tolower(minusc[i]);
}
```

Stampa minuscolo

```
printf("Parola minuscola: %s\n", minusc);
```

Verifica se è palindroma

```
int i, j, palin, lun;
i = 0;
j = strlen(minusc) - 1;
palin = 1;
while(i < j && palin == 1){
    if(minusc[i] != minusc[j])
        palin = 0;
    i++; j--;
}
```



Stampa se palindroma

```
if(palin == 1)
    printf("E' palindroma\n");
else
    printf("Non e' palindroma\n");
```

# Matrici di caratteri

Nel definire una matrice, è ovviamente possibile usare il tipo base **char**

Permette di memorizzare una tabella **NxM** di caratteri ASCII

Ogni posizione **[i][j]** deve contenere un carattere

Non può essere vuota

Non può contenere più di un carattere

```
char tris[3][3] ;
```

	0	1	2
0	O	X	.
1	.	X	.
2	.	.	.

# Vettori di stringhe

Una matrice di caratteri può anche essere vista come:

Un vettore di vettori di caratteri, cioè

Un vettore di stringhe

Si tratta di un metodo diverso di interpretare la stessa struttura dati

```
char nomi[5][10] ;
```

	0	1	2	3	4	5	6	7	8	9
0	F	u	l	v	i	o	\0	x	!	w
1	A	n	t	o	n	i	o	\0	.	Z
2	C	r	i	s	t	i	n	a	\0	u
3	E	l	e	n	a	\0	5	g	r	d
4	D	a	v	i	d	e	\0	\$	2	e

# Vettori di stringhe



Dato un vettore di stringhe, determinare quante volte è presente la lettera 'A' (maiuscola o minuscola)

```
cont = 0 ;
for(i=0; i<MAX; i++) {
    for(j=0; vett[i][j]!='\0'; j++){
        if(toupper(vett[i][j])=='A')
            cont++ ;
    }
}
```

Dato un vettore di stringhe, determinare se esistono due stringhe identiche

```
uguali = 0 ;
for(i=0; i<MAX; i++){
    for(k=i+1; k<MAX; k++){
        if(strcmp(vett[i], vett[k])==0)
            uguali=1 ;
    }
}
```

## Errore frequente



Confondere una stringa (vettore di caratteri) con un vettore di stringhe (matrice di caratteri)

**char** s[LUN+1] ;

- s[i] è un singolo carattere
- s è l'intera stringa

**char** v[MAX][LUN+1] ;

- v[i][j] è il singolo carattere
- v[i] è un'intera stringa
- v è l'intera matrice

# Vettori di stringhe

La stampa del contenuto di un vettore di stringhe si ottiene semplicemente stampando ciascuno degli elementi

Si può utilizzare **puts** oppure **printf**

```
for(i=0; i<N; i++){  
    puts(vett[i]);  
}
```

```
char vett[MAX][LUN+1];  
char s[250];  
...  
do {
```

```
    printf("Quante stringhe? ");  
    gets(s);  
    N = atoi(s);
```

```
    if(N<1 || N>MAX)  
        printf("Valore errato: deve essere  
            tra 1 e %d\n", MAX);  
    } while(N<1 || N>MAX);
```

Acquisire da tastiera un vettore di Stringhe. Un ciclo per ciascuna delle stringhe da leggere. Lunghezza nota a priori  
Lunghezza determinata dalla lettura di un certo dato (es.: "FINE")

```
N = 0;  
end = 0;  
do {  
    printf("Stringa %d: ", N+1);  
    gets(s);  
    if (strlen(s)==0)  
        printf("Vuota - ripeti\n");  
    else if(strlen(s)>LUN)  
        printf("Tropo lunga\n");  
    else if(strcmp(s, "FINE")==0)  
        end = 1;  
    else{  
        strcpy(vett[N], s);  
        N++;  
    }  
} while(end==0);
```

# Esercizio – Vocali e consonanti



Scrivere un programma in linguaggio C che legga una frase introdotta da tastiera.

La frase è terminata dall'introduzione del carattere di invio.

La frase contiene sia caratteri maiuscoli che caratteri minuscoli, e complessivamente al più 100 caratteri.

Il programma dovrà stampare su schermo le seguenti informazioni:

- 1) per ognuna delle lettere dell'alfabeto, il numero di volte che la lettera compare nella stringa
- 2) il numero di consonanti presenti nella stringa
- 3) il numero di vocali presenti nella stringa

# Esercizio – Statistiche sul testo



Un utente inserisce una serie di frasi da tastiera, su più righe

L'inserimento termina quando l'utente inserisce la parola FINE su una riga da sola

Il programma deve determinare:

- 1) Quante righe sono state inserite dall'utente
- 2) Quanti caratteri sono stati inseriti
- 3) Quanti caratteri alfanumerici sono stati inseriti
- 4) Quante parole sono state inserite

```
Prompt dei comandi

Testo: Nel mezzo del cammin di nostra vita
Testo: mi ritrovai per una selva oscura
Testo: che la diritta via era smarrita.
Testo: FINE
L'utente ha inserito 3 righe
L'utente ha inserito 99 caratteri
L'utente ha inserito 82 caratteri alfanumerici
L'utente ha inserito 19 parole
```



# Esercizio – Concorso di intelligenza



In un concorso di intelligenza, **N** giudici esprimono il loro giudizio su **K** candidati.

Il giudizio è un valore numerico tra **0** e **5**

Si scriva un programma in linguaggio C per determinare il candidato più intelligente, ed il giudice più severo

# Esercizio – Concorso di intelligenza



## Azioni:

- 1) Acquisisce il numero di candidati e di giudici
- 2) Acquisisce il voto di ciascun giudice per ciascun candidato
- 3) Somma tutti i voti ottenuti da un candidato
- 4) Somma tutti i voti dati da un giudice
- 5) Il candidato migliore è quello che ha ottenuto il punteggio massimo
- 6) Il giudice più severo è quello che ha dato i voti più bassi (somma)

# Esercizi



In crittografia il cifrario di **Cesare** è uno dei più antichi algoritmi crittografici di cui si abbia traccia storica.

È un cifrario a sostituzione monoalfabetica in cui ogni lettera del testo in chiaro è sostituita nel testo cifrato dalla lettera che si trova un certo numero di posizioni dopo nell'alfabeto.

Scrivere un programma che prenda in input una **stringa** ed un valore intero **N**. Il programma esegue le seguenti operazioni:

- 1) Cifra il testo sostituendo ogni lettera in posizione **i** nell'alfabeto con la lettera in posizione  **$i+N$**
- 2) Stampa la stringa cifrata
- 3) Decifra la stringa codificata con la sostituzione inversa
- 4) Stampa la stringa in chiaro

