



Linguaggio C++

Stringhe in stile C

Definizione ed esempi

- In C una stringa è un vettore di caratteri che termina con il carattere `'\0'`.

- `char saluto[4]={ 'c', 'i', 'a', 'o' };` //E' un vettore di catteri, non una stringa!

- `char saluto[] = "ciao";` //E' una stringa! Il carattere `'\0'` viene
//aggiunto automaticamente dal compilatore

Quanto vale? $4 + 1 = 5$

Esempi

■ `char saluto[5]={‘c’,’i’,’a’,’o’,’\0’};` //E’ una stringa!

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

`char str[5] = “”;` //doppi apici, doppi apici: senza blank

\0	?	?	?	?
----	---	---	---	---

`char frase [] = “Analisi dei requisiti”;` //comodo non dover indicare la dimensione



Operazioni di I/O di stringhe

```
char nome[21];  
    cout<<"Introduci il tuo nome"<<endl;  
    cin>>nome;           //numero massimo di caratteri 21-1=20 perché viene poi  
                           //aggiunto automaticamente il carattere '\0'  
    cout<<"Salve "<<nome<<endl;  
    .....
```

Note:

- 1) La lettura si interrompe quando si incontra il carattere blank (spazio), il carattere di tabulazione o il carattere di ritorno a capo.
- 2) Problema con il buffer di tastiera (pulizia del buffer di tastiera).



I/O di stringhe

- Per ovviare a questi inconvenienti si può ricorrere al metodo

- `cin.getline(varStr, dimVarStr);`

Esempio:

```
char nome[21];  
    cin.getline(nome,21);
```



I/O di stringhe

- In particolare le operazioni che vengono eseguite dal sistema quando viene incontrata questa funzione sono:
 1. Viene estratta dal buffer di input una sequenza di caratteri fino ad incontrare il delimitatore '\n' oppure fino al raggiungimento della dimensione massima specificata, diminuita di uno. Il carattere delimitatore non viene inserito nella stringa;
 2. Viene aggiunto in coda alla sequenza di caratteri estratti, il delimitatore '\0';
 3. La sequenza così completata viene memorizzata nella variabile specificata (1° parametro).



I/O di stringhe

- Per la pulizia del buffer di input:

`cin.ignore(lunghezzaBufferInput, '\n');`



I/O di stringhe

- Nessun problema nella stampa:

```
cout<<str;
```




Attenzione!!!

$1 \neq '1' \neq "1"$



Esempi

```
char stringa[7];  
int i;  
    for(i=0; i<7; i++)  
        stringa[i] = 'A' + i;
```

↑
Array di caratteri

```
char stringa[7];  
int i;  
    for(i=0; i<6; i++)  
        stringa[i] = 'A' + i;  
stringa[i]='\0';
```

↑
Stringa



Costanti

- `#define SALUTO "Ciao"` //direttiva al preprocessore
- `const char SALUTO [] = "Ciao";` //istruzione C/C++



Osservazioni

- Come per i vettori, non viene effettuato alcun controllo sui limiti (bounds checking).
- Visto che le stringhe sono dei vettori di caratteri, è possibile riferirsi ad ogni carattere singolarmente: `str[i]`
- Il carattere di fine stringa (`'\0'`), avendo codice ASCII zero, ha valore di falso.
- Non è possibile fare assegnazione diretta, né confronti fra stringhe (si devono usare funzioni di libreria).



Alcune funzioni di libreria

```
#include <string.h>
```

strlen(); //ritorna il numero di caratteri di una stringa (non conta il carattere di fine stringa)

Esempio:

```
int main(void) {  
    char string[ ] = "Noi siamo i migliori!";  
  
    cout<<strlen(string);  
    return 0;  
}
```



■ **strcpy();**

//per effettuare l'assegnazione fra stringhe

Nota: strcpy(destinazione, sorgente);

Esempio:

```
int main(void) {  
char str[10] = "abcdefghi";  
char str2[10] = "Paolo";
```

```
    strcpy(str, str2);  
    cout<<str<<endl;  
    return 0;  
}
```



■ **strcat();**

//concatena due stringhe

Nota: strcat(primaStr,secondaStr); //concatena alla primaStr la secondaStr

Esempio:

```
int main(void) {  
    char destination[25];  
    char blank[2] = " ";  
    char str[ ] = "a tutti!";  
  
    strcpy(destination, "Ciao");  
    strcat(destination, blank);  
    strcat(destination, str);  
  
    cout<<destination<<endl;  
    return 0;  
}
```



■ **strcmp();** //per confrontare due stringhe (maiuscole ≠ minuscole)

Nota: `esito= strcmp(s1,s2);`

Return Value

If s1 is...	return value is...
-------------	--------------------

less than s2	< 0
--------------	-----

the same as s2	== 0
----------------	------

greater than s2	> 0
-----------------	-----



Esempio:

//conversione di una stringa in maiuscolo

```
char nome[21];  
    strcpy(nome, "Paolo");  
    for(i=0; nome[i]; i++)  
        nome[i] = toupper(nome[i]);    //<ctype>  
    cout<<nome<<endl;
```



//conversione di una stringa in maiuscolo

```
char nome[21];  
    strcpy(nome, "Paolo");  
    strupr(nome);                //strlwr()  
    cout<<nome<<endl;
```



Altre funzioni

#include <stdlib>

- atoi(); //converte una stringa in intero
- atol(); //converte una stringa in intero (long)
- atof(); //converte una stringa in double

num = atoi('1'); NO!!!

num = atoi("1"); SI!!!



Funzioni per la manipolazione di caratteri

#include <ctype>

- `int isalnum(int car);` //restituisce $\neq 0$ se l'argomento è una lettera dell'alfabeto o un numero
- `int isalpha(int car);` //restituisce $\neq 0$ se l'argomento è una lettera dell'alfabeto
- `int isdigit(int car);` //restituisce $\neq 0$ se l'argomento è una cifra
- `int islower(int car);` //restituisce $\neq 0$ se l'argomento è una lettera minuscola
- `int isupper(int car);` //restituisce $\neq 0$ se l'argomento è una lettera maiuscola

- `int tolower(int car);` //converte in minuscolo
- `int toupper(int car);` //converte in maiuscolo



Ancora ... inizializzazioni

- `char str[6] = "salve";` `//stringa`
- `char str2[6] = {'s','a','l','v','e','\0'};` `//stringa`
- `char str3[6] = {'s','a','l','v','e'};` `//array di char`

- `char msg1[] = "Valore introdotto non corretto\n";`



Vettori di stringhe

- Per creare un vettore di stringhe ci si serve di una matrice di caratteri dove la dimensione dell'indice di sinistra determina il numero di stringhe, mentre la dimensione dell'indice di destra specifica la dimensione di ciascuna stringa.

char mesi[12][10];

12 → numero di stringhe (i mesi dell'anno)

10 → ogni stringa è lunga max 10 (9 caratteri effettivi + '\0')



Operazioni di I/O

```
char mesi[12][10];
```

```
...
```

```
cin>>(mesi[i]);                                //i-esima stringa del vettore
```

```
...
```

```
cout<<mesi[i];                                //i-esima stringa del vettore
```

```
...
```

NOTA: Ogni riga della struttura dati (vettore di stringhe/matrice) è una stringa.



NOTA BENE

```
#define MAX_MESI 12
```

```
#define LUNG_MESE 9+1
```

```
char mesi[MAX_MESI][LUNG_MESE];
```

La variabile “mesi” così com'è definita può essere:

- a) matrice di caratteri
- b) vettore di stringhe

sarà l'una o l'altro a seconda di come verrà utilizzata.



Inizializzazione

```
char mesi[MAX_MESI][LUNG_MESE] = {"Gennaio",  
                                     "Febbraio", "Marzo", ... "Dicembre"};
```

Nota:

Ovviamente non si possono mettere i "puntini", i valori devono essere indicati tutti.