



LAVORARE CON LE STRINGHE

*Operazioni elementari sulle stringhe
in C/C++*

Roberto Nardone, Luigi Romano

- Lavorare con le stringe in C
 - Funzioni di copia, concatenazione, conversione, ...
- La «classe» string
 - Operazioni con le stringhe in C++
- Esercizi

- ❑ In C non esiste un tipo predefinito in grado di memorizzare stringhe di caratteri.
- ❑ Una stringa è definita come un array di **char** con terminatore **NULL** e si parla di **stringhe stile C (C-style string)**.
- ❑ Un array di caratteri però non è un tipo di dato e quindi non è possibile applicare su di esso gli operatori definiti dal linguaggio C.
- ❑ Ad esempio non è possibile usare l'operatore **+** per **concatenare** due stringhe stile C.

- ❑ Consideriamo il seguente array di caratteri;

```
char s[10];
```

- ❑ In s sarà possibile memorizzare 9 caratteri (l'ultimo è riservato al carattere di terminazione).
- ❑ Il tentativo di memorizzare una stringa di caratteri con lunghezza maggiore di 9 darà luogo ad un errore di **segmentation fault**.
- ❑ L'inizializzazione di un array di caratteri è analoga a quella di un array di qualsiasi altro tipo e può avvenire come segue:

```
char mystring[] = {'H','e','l','l','o','\0'};  
char mystring[] = "Hello";
```

- ❑ Assegnazioni del tipo `mystring[] = "Hello"` non sono consentite, in quanto l'assegnazione è ammessa solo al momento della dichiarazione dell'array.
- ❑ Per assegnare una stringa di caratteri ad un array precedentemente dichiarato, si devono fare tante assegnazioni quanti sono i caratteri dell'array:

```
mystring[0]='H';  
mystring[1]='e';  
mystring[2]='l';  
mystring[3]='l';  
mystring[4]='o';  
mystring[5]='\0';
```

- ❑ Per semplificare le operazioni sulle stringhe, la libreria standard mette a disposizione un set di funzioni raggruppate nell' header file **cstring**.
- ❑ Di seguito alcune funzioni fornite dalla libreria per manipolare le stringhe(*):
 - **strcpy(s1, s2)**: Copia la stringa s2 in s1 e ritorna s1;
 - **strcat(s1, s2)**:Accoda la stringa s2 alla stringa s1 e ritorna la stringa concatenata;
 - **strlen(s1)**:Ritorna la lunghezza della stringa s1;
 - **strcmp(s1, s2)**:Confronta le due stringhe e ritorna 0 se sono uguali.

* <http://www.cplusplus.com/reference/cstring/>

- ❑ La funzione **strcpy** consente la copia di una stringa in un'altra stringa di dimensione tale da contenere la stringa di partenza.
- ❑ Il prototipo della funzione definito nell'header file **cstring** è il seguente:

char * strcpy(char* destination, const char* source);

- ❑ La funzione copia la C-string puntata da **source** nell'array puntato da **destination**, compreso il carattere di terminazione.

LA FUNZIONE STRCPY: ESEMPIO D'USO

8

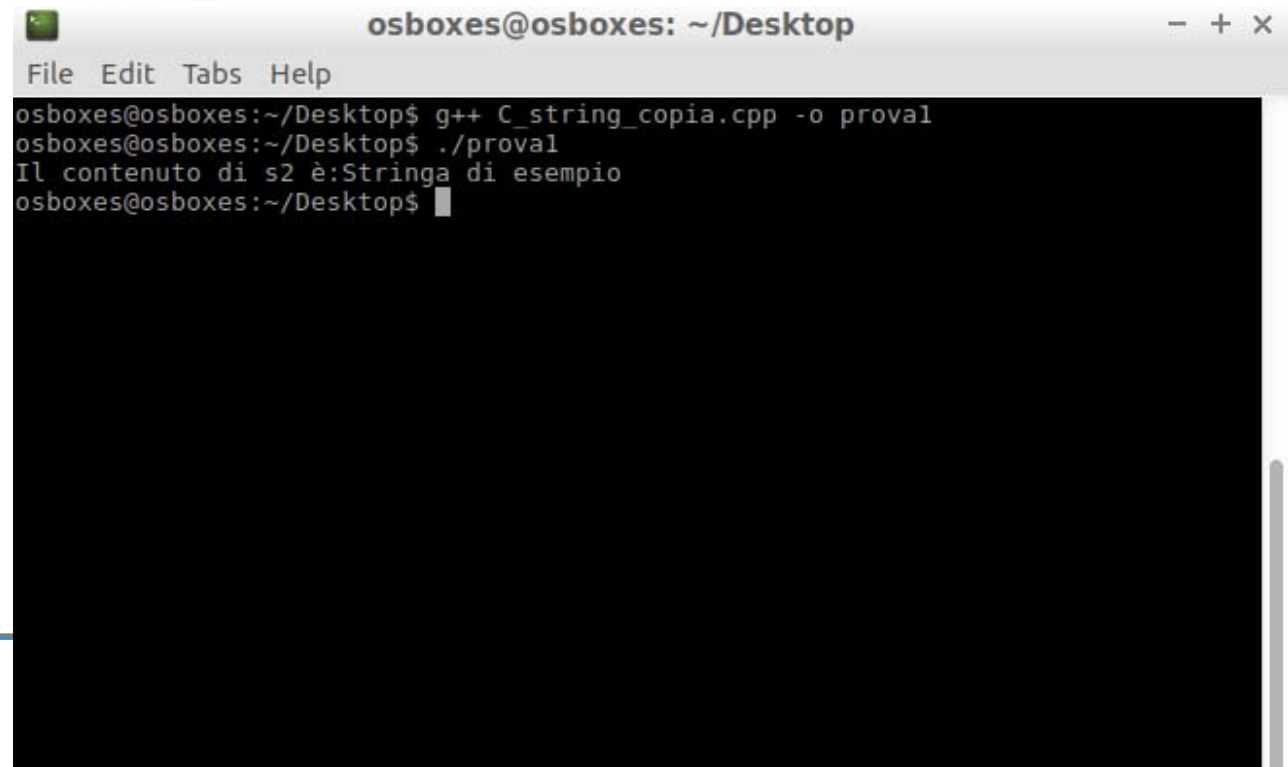
```
#include<iostream>
#include<cstring>

using namespace std;

int main()
{
    char s1[]="Stringa di esempio";
    char s2[40];

    strcpy(s2, s1);

    cout<<"Il contenuto di s2 è:"<<s2<<endl;
    return 0;
}
```



A terminal window titled "osboxes@osboxes: ~/Desktop" with standard window controls. The terminal shows the following commands and output:

```
osboxes@osboxes:~/Desktop$ g++ C_string_copia.cpp -o prova1
osboxes@osboxes:~/Desktop$ ./prova1
Il contenuto di s2 è:Stringa di esempio
osboxes@osboxes:~/Desktop$
```


- ❑ La funzione **strcmp** confronta due stringhe secondo l'ordinamento lessicografico restituendo il valore intero che indica la relazione fra le due stringhe.
- ❑ Il prototipo della funzione definito nell'header file è:

int strcmp (const char* str1, const char* str2);

- ❑ La funzione restituisce:
 - valore 0 se le due stringhe sono **uguali**;
 - valore <0 se la prima stringa **precede** la seconda;
 - valore >0 se la prima stringa **segue** la seconda.

*Scrivere un programma che a partire da un array di caratteri inizializzato con il nome di una persona consenta, nota la dimensione dell'array(vedi la funzione **strlen**), di indovinare il nome memorizzato.*

```
#include<iostream>
#include<cstring>

using namespace std;

int main()
{
    char nome[] ="Giovanni";
    char input[40];
    cout<<"La lunghezza del nome da indovinare è: "<<strlen(nome)<<endl;

    do{
        cout<<"Inserire un nome:"<<endl;
        cin>>input;
    }while(strcmp(nome,input) !=0);

    cout<<"Risposta Esatta!!!"<<endl;
    return 0;
}
```

```
char* strcat(char* destination, const char* source);
```

- ☐ La funzione **strcat** concatena una copia della stringa **source** alla stringa **destination**.
- ☐ Il carattere di terminazione della stringa **destination** viene sovrascritto dal primo carattere della stringa **source**.
- ☐ Il carattere di terminazione viene incluso alla fine della stringa **destination**.

USO DELLE FUNZIONI DELLA LIBRERIA

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

#define MAX_DIM 50

using namespace std;

int main()
{
    char s1[] = "Hello";
    char s2[] = "World";
    char s3[MAX_DIM]={};

    cout<<"La lunghezza della stringa s1 è: "<<strlen(s1)<<endl;
    cout<<"-----"<<endl;

    strcpy(s1, s2);

    cout<<"Dopo l'operazione di copia, il contenuto di s1 è: "<<s1<<endl;
    cout<<"-----"<<endl;

    cout<<"La comparazione fra le stringhe s1 ed s2 è: "<<strcmp(s1, s2)<<endl;
    cout<<"-----"<<endl;

    strcat(s3, "These ");
    strcat(s3, "Strings ");
    strcat(s3, "are ");
    strcat(s3, "concatenated.");
    cout<<s3<<endl;
    cout<<"-----"<<endl;

    return 0;
}
```

```
osboxes@osboxes: ~/Desktop/Esecizi Array
File Edit Tabs Help
osboxes@osboxes:~/Desktop/Esecizi Array$ ./complete
La lunghezza della stringa s1 è: 5
-----
Dopo l'operazione di copia, il contenuto di s1 è: World
-----
La comparazione fra le stringhe s1 ed s2 è: 0
-----
These Strings are concatenated.
-----
osboxes@osboxes:~/Desktop/Esecizi Array$
```

- ❑ La **cstdlib** mette a disposizione una serie di funzioni per convertire una stringa in un valore numerico.

double atof(const char* str);

int atoi(const char* str);

long int atol(const char* str);

long long int atoll(const char* str);

- ❑ La funzione **atof (alpha to float)** interpreta la stringa **str** come un numero a virgola mobile (**floating point**) e ritorna il suo valore come **double**.

- ❑ Le funzione **atoi**, **atol** e **atoll** interpretano la stringa **str** in ingresso come un numero e ritornano rispettivamente un **int**, un **long int** e un **long long int**.

- ❑ Le funzione **atoi**, **atol** e **atoll** interpretano la stringa **str** in ingresso come un numero e ritornano rispettivamente un **int**, un **long int** e un **long long int**.

```
#include<iostream>

using namespace std;

int main()
{
    char str_number[] = "-189.2";

    int str_to_int = atoi(str_number);

    double str_to_double = atof(str_number);

    long int str_to_long = atol(str_number);

    long long int str_to_long_long = atoll(str_number);

    cout<<"La conversione della stringa in intero è: "<<str_to_int<<endl;
    cout<<"-----"<<endl;
    cout<<"La conversione della stringa in double è: "<<str_to_double<<endl;
    cout<<"-----"<<endl;
    cout<<"La conversione della stringa in long int è: "<<str_to_long<<endl;
    cout<<"-----"<<endl;
    cout<<"La conversione della stringa in long long int è: "<<str_to_long_long<<endl;

    return 0;
}
```


- ❑ Le funzione **atoi**, **atol** e **atoll** interpretano la stringa **str** in ingresso e restituiscono rispettivamente un **int**, un **long int** e un **long long int**.

```
osboxes@osboxes: ~/Desktop
File Edit Tabs Help
osboxes@osboxes:~/Desktop$ g++ Conversione.cpp -o es4
osboxes@osboxes:~/Desktop$ ./es4
La conversione della stringa in intero è: -189
-----
La conversione della stringa in double è: -189.2
-----
La conversione della stringa in long int è: -189
-----
La conversione della stringa in long long int è: -189
osboxes@osboxes:~/Desktop$
```

```
cout<<"La conversione della stringa in long int è: "<<str_to_long<<endl;
cout<<"-----"<<endl;
cout<<"La conversione della stringa in long long int è: "<<str_to_long_long<<endl;

return 0;
}
```


- `int isdigit(int c);` riceve un carattere e ritorna 1 se corrisponde ad una cifra 0 in caso contrario
- `int isalpha(int c);` riceve un carattere e ritorna 1 se corrisponde ad una lettera 0 in caso contrario
- `int isspace(int c);` riceve un carattere e ritorna 1 se corrisponde ad un carattere di spazio (spazio, tab, newline, ...) 0 in caso contrario
- `int islower(int c)` e `int isupper(int c)` controllano se il carattere è minuscolo o maiuscolo
- `int tolower(int c)` e `int toupper(int c)` ritornano il corrispondente minuscolo o maiuscolo di c

- ❑ Scrivere un programma che data una stringa la converta in maiuscolo
- ❑ Scrivere un programma che data una stringa estragga le cifre in essa contenute
- ❑ Scrivere un programma che faccia il trim sinistro e destro di una stringa (tolga gli spazi vuoti da sinistra e destra... es. “ Luigi Coppolino ” diventa “Luigi Coppolino”
- ❑ Scrivere un programma che legga nome cognome e anno di nascita dell'utente e risponda con un messaggio del tipo “ciao nome cognome, hai x anni” dove x è calcolato come differenza dall'anno corrente
- ❑ Scrivere un programma che legga dall'utente la sua matricola che può essere nella forma solo numero (es. 327001) oppure preceduta da MATR. (es. MATR.327001) e ritorni in output il corso di laurea indicato dai primi tre caratteri (es. 327) e la matricola (es. 001)



Le stringhe in C++

La classe string

- Aggiungere l'header file `<string>`
- Definizione di una stringa:
 - `string nomestringa`
- Assegnazione: avviene mediante l'operatore `=`
`string a;`
`a = "prova";`
- Concatenazione: avviene mediante l'operatore `+`
`string a = "Luigi";`
`string b = "Coppolino";`
`string c = a+b;`
- Gli operatori di confronto lavorano in maniera intuitiva assumendo un ordinamento di tipo alfabetico (cioè $a < b < c \dots$)
 - `==` `!=` `<` `>` `<=` `>=`

- Il concetto formale di metodo verrà spiegato successivamente => operazione su una stringa...
 - `.size()` o `.length()` : ritorna la lunghezza (escluso terminatore)
 - `.substr(x,y)`: ritorna la sottostringa dal carattere x-esimo per un totale di y caratteri
 - `.c_str()`: ritorna una stringa stile C

□ Es.

```
string a = "pippo";  
cout << a.size(); //ritorna 5  
string b = a.substr(0,3); // b = "pip"  
char c[] = a.c_str();
```

- Si usano gli oggetti cin e cout

```
string a;
```

```
cin >> a;
```

```
cout << a;
```

- Se l'input contiene spazi viene interrotto al primo spazio =>
usare la funzione `getline(cin, stringa_da_leggere)`

- Es.

```
string a;
```

```
getline(cin, a); // inserire "Luigi Coppolino"
```

```
cout << a; // ritorna "Luigi Coppolino"
```

- Rifare gli esercizi precedenti usando la classe string anche cercando nuovi metodi di string da <http://www.cplusplus.com/reference/string/>

- Rifare gli esercizi precedenti usando la classe string anche cercando nuovi metodi di string da <http://www.cplusplus.com/reference/string/>
 - Tips: vedere s to i